COSC460 Honours Report: A Sequential Steady-State Detection Method for Quantitative Discrete-Event Simulation

December 1, 2012

Adam Freeth

afr58@uclive.ac.nz

Department of Computer Science and Software Engineering

University of Canterbury, Christchurch, New Zealand

Supervisors: Professor Krzysztof Pawlikowski and Associate Professor Don McNickle

krys.pawlikowski@canterbury.ac.nz / don.mcnickle@canterbury.ac.nz

Abstract

In quantitative discrete-event simulation, the initial transient phase can cause bias in the estimation of steady-state performance measures. Methods for detecting and truncating this phase make calculating accurate estimates from the truncated sample possible, but no methods proposed in the literature have proved to work universally in the sequential online analysis of output data during simulation. This report proposes a new automated truncation method based on the convergence of the cumulative mean to its steady-state value. The method uses forecasting techniques to determine this convergence, returning a truncation point when the cumulative mean time-series becomes sufficiently horizontal and flat. Values for the method's parameters are found that adequately truncate initialisation bias for a range of simulation models. The new method is compared with the sequential MSER-5 method, and shows to detect the onset of steady-state more effectively and consistently for almost all simulation models that are tested. This rule thus appears to be a good candidate as a robust sequential truncation method and for implementation in sequential simulation research packages such as *Akaroa2*.

Acknowledgements

I would like to wholeheartedly thank my supervisors Krzysztof Pawlikowski and Don McNickle for their encouragement and valuable feedback throughout the duration of this project.

Contents

1 Introduction

1.1 Quantitative Discrete-Event Simulation

In practice, system performance analysis is often not possible using real-world implementations or theoretical models. Implementation and scientific observation in a controlled environment of real-world systems can be too costly (if not impossible), and systems are generally too complex to be analytically tractable. Computer simulation of such systems is a viable alternative for performance analysis, and can be carried out in reasonable time and without significant expense.

The credibility of scientific simulation is dependent upon two general conditions: validity of the simulation model, when the model accurately corresponds to the real system to an appropriate level of detail; and validity of the simulation experiment, when the software implementation is verified to conform the model. Simulation experiments are frequently stochastic—that is, they require sources of randomness to generate data—and these sources must be suitable to ensure the validity of the experiments. As stochastic simulation is not deterministic, it requires particular methods of analysing output data that ensure the integrity of its results, [31].

Quantitative discrete-event simulation is a type of simulation evolving along a sequence of instantaneous events. Measurements are collected throughout the simulation and analysed to estimate values that characterise the performance of the simulation system, contrasting qualitative simulation. It is commonly used for the simulation of state-changing models such as telecommunications networks or inventory systems. The output data of these systems consist of observations that are observed either at discrete equal time intervals or at the occurrence of a particular event.

The type of analysis required for output data of a given simulation depends on whether the simulation is *terminating* or *non-terminating*. Terminating simulations are those that only need to model a system's characteristics within a specific time period, such as a model of a job shop over a single day from its opening to its closing hours. Here, when estimating performance measures of the system, we are only concerned with the values from its beginning to its termination. Non-terminating simulations represent the opposite case: when there is no particular end to the simulation process and it could theoretically continue *ad infinitum*. An example of this is a simulation to determine the long-run performance of a peer-to-peer network.

1.2 Steady-State Simulation

Non-terminating simulation models are either *stable* or *unstable*. A model is stable if it approaches *steady-state*, when the expected distribution of its observation values becomes time-invariant. Alternatively, if a given model never approaches time-stationarity, it is unstable. Steady-state performance measures of stable simulation systems are estimated from the output data, and these performance measures tell us the expected time-independent behaviour of the system in the long run.

A stable system is not necessarily in steady-state at the beginning of the output data. For instance, it could be initialised with conditions that are unrepresentative of steady-state values. Simulation output is generally autocorrelated—when the correlation between two data points in the output is a function of the distance separating them—so unrepresentative initial conditions bias a number of observations at the beginning of the output from steady-state values. These initial observations gradually converge towards steady-state properties and are thus non-stationary, and are termed the *initial transient phase*, [5].

It is only possible to estimate steady-state parameters from an observation sample of finite length in practice. This means that if the observations in the initial transient phase are included in the estimation of these parameters, the resulting estimates can be significantly biased, [4].

Steady-state characteristics are generally not known in advance, because determining them is usually the purpose of steady-state simulation. Thus, simulation practitioners commonly cannot initialise the simulation in steady-state and must use arbitrary initial conditions, meaning that an initial transient phase is probably present in the output data. The presence of an initial transient phase must therefore be taken into consideration to obtain unbiased estimates. The most common approach to mitigate the effect of the initial transient phase is to delete these observations from the observation sample, and estimate the performance measures from the remaining truncated sample.

1.3 Sequential Steady-State Analysis

Sequentially analysing simulation systems means that further output data are sequentially collected and analysed until a prespecified level of precision is achieved in the performance measure estimation, [7]. The fast computational speeds of modern computers has made this feasible, and generating large numbers of observations for many simulations is no longer time-consuming. Using sequential analysis, methods that truncate the initial transient phase are not detrimental to the precision of final estimates, because further observations can always be generated and included in the sample.

Akaroa2 is a research tool developed by the Simulation Research Group at the University of Canterbury, designed to assist simulation practitioners in both the sequential analysis of discrete-event simulation and *multiple replications in parallel* (MRIP), [26]. As of November 2012, it has over 5,000 registered users from around the world. As part of its sequential analysis functionality, the stopping criteria are specified by the user, which are maximum allowable values for relative and absolute errors. The simulation stops once the error of the estimate—given by the confidence interval

half-width—falls below either threshold.

Akaroa2 provides a sequential truncation method based on a heuristic rule known as "25 crossings of the mean" and a statistical test developed by Schruben *et al.* [34], but adapted for sequential analysis, [8, 13, 30]. However, Schruben's method occasionally fails to give appropriate truncation points for some simulation output data, shown by recent research performed by the Simulation Research Group. Thus, *Akaroa2* is in need of more robust sequential truncation method: one that can guarantee accurate final point estimates for a comprehensive range of simulation output processes. No such method is currently known in the simulation literature.

1.4 Aim and Objectives

The aim of this research project was to develop a robust sequential method for detecting the onset of steady-state. Focussing on the detection of the onset of steady-state rather than the end of the initial transient phase helps to ensure that the beginning of the truncation sample is sufficiently in steady-state, so that accurate estimates are found from the remaining data. A method that aims to detect the end of the transient phase rather than the onset of steady-state does not necessarily guarantee that observations that follow the truncation point represent steady-state.

The method developed in this research is based on the convergence of the mean of the process (and possibly other performance measures) to its steady-state value, as the number of observations t increases, $t \to \infty$, [5]. This is similar to the reasoning behind the popular Welch's method, [36]. Specifically, it detects the onset of steady-state at the point that the cumulative mean time-series becomes suitably flat, where the cumulative mean at point t is the mean of all observations up to point t.

The method needed to be established as a robust technique and should sufficiently mitigate initialisation bias across arbitrary simulation output. The developed method is subjected to comprehensive testing across a wide variety of simple simulation models, to establish its effectiveness as a general-purpose rule. Various combinations of the method's parameters are tested to determine those most suitable. The method is implemented in *Akaroa2* to allow it to truncate observations up to a point that lies in or sufficiently near steady-state, ensuring that final point estimates are unbiased for arbitrary simulation.

The structure of the remainder of this report is as follows. Chapter 2 overviews the problem of the initial transient phase and methods proposed to mitigate its effect. The new truncation method is detailed in Chapter 3, and an investigation into suitable parameters for this method is shown in Chapter 4. Using these parameters, the method is then evaluated in comparison to the sequential MSER-5 method in Chapter 5. A discussion of the findings and limitations of the project are given in Chapter 6, and finally Chapter 7 summarises the outcomes and achievements of the project.

2 Initial Transient Phase

2.1 Mitigating Initialisation Bias

Although simulation practitioners usually do not know all steady-state characteristics of the simulation model *a priori*, they may know some aspects of it. In these cases, it is tempting to simply initialise the simulation with the known conditions, such as with the mean or the most probable state of the process. For example, Grassman [15] recommends that the simulation should be initialised in its most probable state when estimating performance measures. Both Kelton [19] and Abate and White [1] suggest that some queueing models converge to their steady-state means fastest when initialised at approximately one-and-a-half times these means. As the actual steady-state mean is generally unknown in advance in real-world simulations, Pawlikowski [30] recommends that systems should be initialised empty and idle, because systems that are initialised from highly underestimated values generally converge to steady-state faster than systems initialised from highly overestimated values.

Despite the ability to reduce the effect of the initial bias through intelligent selection of initial conditions, an initial transient phase can still be present. Also, in many scenarios, *a priori* knowledge of any steady-state characteristics are simply not known. Thus, certain techniques are needed to ensure that initial bias in the output is mitigated as much as possible. Novel approaches such as the use of *simulation slithers* have not yet been developed to a sufficient degree for use in practice on arbitrary simulation models, [2].

2.2 Truncating the Initial Transient Phase

A popular and effective method to mitigate initialisation bias is to determine a point where the initial transient phase appears to end and the steady-state begins, discarding the prior observations and analysing only the remaining sample, [25]. The point where this occurs is the *truncation point*. However, identifying such a point is not trivial, as different simulation models can converge to steady-state in diverse ways, and not necessarily monotonically, [6]. A truncation point is considered valid when it lies sufficiently close to or within the steady-state, such that there is little or no initialisation bias remaining in the observations following it. Conversely, an invalid truncation point is one that does not properly truncate the initial transient phase, when nontrivial levels of bias remain in the truncated sample.

A variety of requirements have been proposed for determining whether a given truncation point

is valid or not. Pawlikowski [30] summarises a number of rules-of-thumb, which can also be implemented as truncation methods. These are just heuristics and cannot give any guarantees as to how much initial bias has been eliminated.

White [38] suggests that it is suitable for truncation to occur at the point where the most common observation value is outputted. However, simulations quite often have very large or infinite numbers of possible output values, so it is possible for even the most common value to occur very infrequently, and it can exist far away from mean of the output process. Therefore, systematic choice of such a truncation point can lead to bias in final estimates. A similar consequence can occur when other specific output values are chosen as the truncation point. For example, systematically identifying truncation points at observation values near the mean of the process can reduce variance in the following observations from the expected steady-state variance.

Another possible requirement is the point where the *mean squared error* (MSE) of the resulting truncated sample is minimised, [29]. The mean squared error of an estimator $\hat{\theta}$ can be defined as

$$MSE(\hat{\theta}) = Var(\hat{\theta}) + (Bias(\hat{\theta}, \theta))^2,$$

where θ is the system parameter being estimated. Mean squared error is thus a combined measure of variation and systematic bias in estimates on the truncated sample. Variance can usually be reduced by increasing the sample size, as this increases the precision of the estimator. However, including further observations from the initial transient phase can induce bias, as these observations are not representative of steady-state values. Thus, for fixed sample sizes, there is no clear truncation point for obtaining accuracy and precision. Minimising the MSE, as it is a combination of both bias and variance, will just give a point that compromises between minimising bias and minimising variance. Currently, the bias of the final estimators is a more popular measure of quality of final results than MSE, thus the deletion of atypical observations are of significant concern. This is particularly true in the case of sequential analysis, when further observations are successively collected, reducing the variance of the resulting estimator regardless of how many observations were truncated.

In sequential analysis, the range of proposed specifications described above are all lenient when determining what constitutes steady-state. That is, they are unable to give any guarantee of how well the observations following the truncation point will approximate the process's steady-state. This is largely because they are designed for fixed-sample-size analysis, in which case truncating many data can lead to an insufficiently small number of remaining observations to analyse. Sequential analysis does not have this restriction, so more flexible rules can be applied. Eickhoff [5] describes a requirement for the onset of steady-state that is suitable for sequential analysis, given by

$$\forall(i \geq l_F, \Delta \geq 0, x) : F_{X_t}(x) \simeq F_{X_{t+\Delta}}(x), \tag{2.1}$$

where l is a given truncation point, X_t is the t^{th} observation, $F(x)$ is the cumulative distribution function, and "$\simeq$" signifies the closeness of the two distributions. In other words, the onset of steady-state

is given by the point where the distributions of all following points are approximately equal, that is, time-independent. Given a measure of estimating the closeness of two distributions, this gives an appropriate condition for determining whether a truncation method sufficiently detects the onset of steady-state, regardless of the moment of performance measure X_t that is analysed.

The methods proposed so far operate under specific assumptions about stochastic properties of the simulated processes, and most of them have been proposed for non-sequential simulation only. Additionally, no method of detecting the end of the initial transient phase is known that could be applied to any performance measure of the simulated system, such as mean values, variances or quantiles.

2.3 Proposed Truncation Methods

Over 40 truncation methods have been proposed in the literature, [18], most of which apply to offline non-sequential simulation analysis rather than online sequential analysis. Pawlikowski [30] surveys an array of early proposed methods, including heuristic rules based on the rule-of-thumb examples given in Section 2.2. Hoad *et al.* [18] surveys 42 truncation methods, organising them into the following categories:

Graphical methods. Truncation points are determined by visual analysis of a time-series plot of output data.

Heuristic approaches. General, basic rules for determining truncation points, without significant theoretical bases.

Statistical methods. Determining truncation points based upon statistical techniques. including tests that determine the presence of initialisation bias in the output data.

Hybrid methods. Compositions of initialisation bias tests to determine truncation points.

Welch's method [36] is an off-line graphical method that has proven popular due to its simplicity and intuition. A number of independent simulations runs are performed, and a time-series of the mean value obtained at each observation point is plotted. It is expected that averaging over a number of runs will help to distinguish the initial transient phase from steady-state, as the initial transient phase has a non-stationary mean unlike in steady-state. The simulation practitioner then visually observes this time-series to determine the point at which the mean appears to become stationary, and this is the truncation point. A similar off-line method has been adapted to the cumulative mean—or running mean—of the output process, [33]. This allows the time-series to be averaged over a greater number of data, smoothing the time-series and making it easier to determine when it becomes stationary.

2.3.1 Sequential MSER-5

Currently, one of the most popular methods in literature is the *Marginal Standard Error Rule* (MSER), [9, 16–18, 21, 27, 29, 33, 35, 38–40], first proposed as the Marginal Confidence Rule by McClarnon

[22]. This method has even been proposed as the most promising for sequential truncation by Hoad *et al.* [18]. Given a fixed output sample of n observations, a test statistic is calculated at each observation point as

$$MSER(n,l) = \frac{S^2_{n,l}}{n-k},$$

where $S^2_{n,l}$ is the sample variance of the remaining $n-l$ observations, given by

$$S^2_{n,l} = \frac{\sum_{i=l}^{n}(X_i - \overline{X}_{n,l})^2}{n-k}.$$

Here, l is the tentative truncation point, X_t is the value of the t^{th} observation, and $\overline{X}_{n,l}$ is the calculated mean of the last $n-l$ observations. The truncation point is determined by the value of l that gives the minimum $MSER(n,l)$ statistic. As the sample size n approaches infinity, $n \to \infty$, the point of the minimum $MSER(n,l)$ statistic approaches the truncation point that gives a minimum mean squared error of the final estimator $\hat{\theta}$. Thus, for sufficiently large values of n, the MSER method should give truncation points that approximately minimise the mean squared error, [29].

The data used by MSER are often preprocessed by taking batch means of non-overlapping batches of m successive observations. This gives MSER-m, where m is typically 5. This smooths the data to reduce the likelihood of outliers negatively influencing the truncation point determination, and also reduces computational time.

MSER-5 has been adapted as a sequential truncation method, [17]. A number of initial observations n are taken and the MSER statistic is computed over these data. Test statistics for the last c batches are not calculated, as these can give erratic values due to being computed from such a small data set, and this number of data is not sufficient to confidently determine a truncation point regardless. A default value of $c = 5$ is used.

For this sequential version, further prewhitening can be applied by averaging observations over k simultaneous independent simulation runs, such that $k > 1$. This is a similar technique to that used by Welch's method as described above, helping to smooth the data. However, as this research project only concerns sequential simulation in the context of single simulation runs, $k = 1$ is assumed.

Given the number of batches $b = \lfloor n/m \rfloor$ and the point of the minimum MSER statistic l^*, if $l^* \leq (b-c)/2$, then l^* is determined to be a valid truncation point and the rule returns with this value. However, if $l^* > (b-c)/2$, then l^* is deemed invalid, as it may suggest that the entire sample is still in the initial transient phase, and that the steady-state lies beyond it. Thus, further observations are sequentially obtained to create $\lfloor z \times b \rfloor$ more batches, where z defaults to 10%. The MSER statistics are again computed for all data points on this larger sample, and the same conditional is applied to determine if a valid truncation point is found or not. This process of sequentially extending the number of observations and batches continues in this way until a valid truncation point is found.

Due to the way that this sequential method is designed—applying the test to a fixed size of data, then sequentially extending it and reapplying the test whether certain conditions are met—its validity

as an effective truncation method is at least partially dependent upon the validity of the non-sequential version of the method. Specifically, if the non-sequential version incorrectly determines a truncation point in the first half of the data when in fact the entire sequence is within the initial transient phase, the sequential version will also. Freeth *et al.* [10] show that the non-sequential MSER-*m* fails to find an invalid truncation point (that is, one that lies in the second half of the MSER statistics) for some cases when the entire sequence is subsumed in the initial transient, and thus the sequential implementation fails as well. The MSER-*m* method is also found to consistently find no initial transient phase (that is, a truncation point at the first observation) for empty-and-idle M/M/1 queues with traffic intensity of $\rho < 1$.

3 New Steady-State Detection Method

3.1 Convergence of Cumulative Mean

A cumulative mean of a time-series at point t is a running mean of all observations from X_0 to X_t inclusive, producing a new time-series C_t given by

$$C_t = \frac{1}{t+1} \sum_{i=0}^{t} X_i.$$

So at the given point t, C_t is the mean of $t+1$ observations. This time-series C_t thus becomes smoothed as t increases, assuming the observations are taken from a stable system, because each additional observation has a progressively smaller weighting on the value of C_t as the sample size increases.

The ergodic theorem for time-stationary stochastic processes shows that the expected steady-state value $E[X_\infty]$ for process X_t is given by

$$E[X_\infty] = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{+T/2} x_t di = \overline{X}_t,$$

where x_t is a realisation of X_t and $\overline{X}_t$ is an estimator of $E[X_\infty]$. This holds true regardless of whether any initial transient is present in the output, because as the number of observations increases to infinity, the effect of the initial transient becomes negligible. Thus, C_t converges to the steady-state value $E[X_\infty]$ as t increases. In terms of a plot of C_t, the graph becomes flat and horizontal for large t. Figure 3.1 gives examples of the cumulative mean for a number of different output processes, which all converge to their steady-state values as further observations are included, although this rate of convergence varies. We consider the following processes: waiting times of an empty-and-idle M/M/1 queue with traffic intensity $\rho = 0.5$; a geometrical ARMA(1,1) process initialised with $X_{-1} = 0$; an AR(1) process with autoregressive parameter $\varphi = 0.9$ and initial bias $b = 100$; and a damped vibration process with amplitude $k = 10$, period $T = 50$ and initial transient length $l = 1,000$. See Appendix A for further details.

As the mean of the underlying process can be assumed to be in steady-state when the time-series of

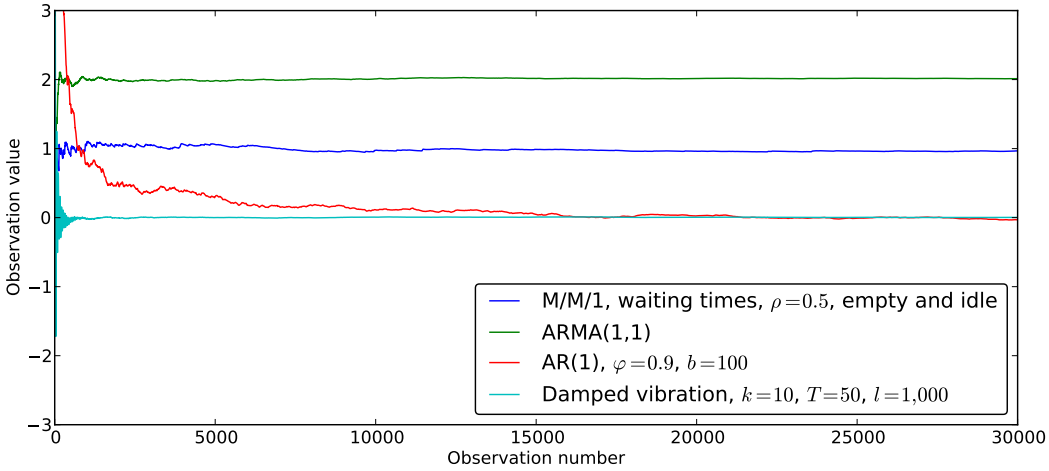


Figure 3.1: Examples of the cumulative means of various processes converging to their steady-state means. The M/M/1 queue has a steady-state mean $E[X_\infty] = 1$, the ARMA(1,1) process has $E[X_\infty] = 2$, and the remaining two processes both have $E[X_\infty] = 0$. See Appendix A for details.

C_t becomes suitably horizontal, a point on this time-series when an appropriate degree of flatness has been reached is used as a truncation point, with the initial transient phase preceding it and steady-state following it. The intuition behind this is similar to that of Welch's method, as discussed in Section 2.3, except that this is applied to a single simulation run as opposed to combining together multiple independent runs.

Thus, analysing the cumulative mean time-series can be used to determine a truncation point. This is also applicable to sequential analysis, because the cumulative mean can be sequentially calculated as further observations are obtained. A method for algorithmically determining the flatness of the cumulative mean time-series needs to be developed to allow for automated truncation point detection.

3.2 Forecasting Methods to Determine Horizontal Flatness

Forecasting methods can be used to automatically decide when the cumulative mean plot has become sufficiently flat and horizontal to give a truncation point. Such methods are used by Mackulak *et al.* [20] on cumulative means to develop a sequential simulation stopping rule. A similar method could be designed for determining a truncation point, which would presumably have relaxed conditions for detecting flatness, as it needs to detect the end of the initial transient phase rather than overwhelm the initial transient by determining the stopping point of a long simulation run.

Linear regression analysis is one possible method of forecasting that could be used to determine flatness, where flatness is given by the slope of the regression line along with the coefficient of determination. However, in preliminary tests, this displayed highly disparate types of behaviour across varying simulation models, so it appears to be unsuitable as a universal method for determining convergence to steady-state.

Smoothing models are another potential method for identifying when the plot becomes horizon-

tally flat. Single exponential smoothing is used in the method developed by Mackulak *et al.* [20], yet this does not take into account any trend in the data. However, double exponential smoothing does, and there are two possible methods for this: Holt-Winters double exponential smoothing [12] and Brown's linear exponential smoothing, [3]. Both of these methods compute a term for slope of the trend line that can then be used for forecasting future observations. Nonetheless, preliminary tests showed that both of these double exponential methods gave similar erratic behaviour to that of linear regression across different simulation models. However, forecasting using slopes is not necessary (and perhaps not even desirable), as we are trying to predict when the slope becomes horizontal. Forecasts without using slopes will simply become more accurate as the cumulative mean converges to flatness, and poor accuracy while the time-series in non-stationary will only be amplified, potentially allowing for easier determination of when the initial transient ends and steady-state begins.

Preliminary tests showed single exponential smoothing having relatively consistent behaviour across varying simulation models, so it is used as the forecasting method for deciding horizontal flatness in this report. The smoothed time-series s_t is given recursively by

$$s_t = \alpha C_t + (1-\alpha)s_{t-1}, \qquad t \geq 1,$$

where $s_0 = C_0$ and α is the smoothing factor such that $0 < \alpha < 1$. Lower values for α will induce more smoothing, that is, the smoothed value will be less weighted toward the recent cumulative means and more heavily weighted to values earlier in the sequence. The value of s_t can then be used as forecasts for subsequent cumulative means C_{t+i} for $i > 0$.

As new observations are sequentially collected and new cumulative means are computed, the accuracy of the forecasts given by previous smoothed values can be established. One-step-ahead forecasting errors e_t are calculated by the difference between the previous smoothed value and the current cumulative mean, $e_t = s_{t-1} - C_t$. The values for e_t should converge to zero as s_t converges to C_t, which is expected as C_t approaches its steady-state value. These one-step-ahead errors can therefore be used as a guide for detecting the convergence of the cumulative mean C_t. It is not sufficient for single or few values of e_t to approach zero, however, as s_t and C_t could cross over each other due to randomness in the sequences, giving some small values for e_t even within the initial transient phase. A method is needed for detecting when the forecasting errors e_t have become consistently small, so that it can be assumed that s_t has converged close to C_t, with C_t becoming flat.

3.3 Detection Condition

Mackulak *et al.* [20] use a sliding window of N observations that moves along the sequence as further observations are collected (and as C_t, s_t and e_t are calculated). The absolute one-step-ahead forecasting errors $|e_t|$ within this window are summed, giving

$$E_t = \sum_{i=0}^{N-1} |e_{t-i}|, \qquad t \geq N-1. \tag{3.1}$$

Absolute errors are used to ensure that processes that do not converge monotonically to steady-state are also taken into account; specifically, when positive and negative errors would otherwise cancel out one another. This sum E_t is compared to a stopping value proportional to the current cumulative mean C_t. If the summation falls below this stopping value, the simulation stops and the cumulative mean is returned as a final estimate of the steady-state mean. This is intended to give final accuracy of the estimated mean relative to its size, but this is not applicable to detection of the initial transient phase and the onset of steady-state, nor does it ensure a specified maximum statistical error of the final estimates. The size of the initialisation bias and length of the initial transient phase are not necessarily dependent upon the relative mean of the process. Using such a stopping value would probably give underestimated truncation points for processes with large means (due to a relaxed detection condition) or overestimated truncation points for small means (due to a stricter condition). Truncation points may never be found for processes with means of zero. Thus, an alternative detection condition is needed in the case of steady-state detection.

A detection condition based on the sample standard deviation of the forecasting errors, S_e, presents a promising alternative. In this case, the stopping condition compares E_t with the variation in the observed data. The detection condition is thus

$$E_t \leq \gamma N S_e, \tag{3.2}$$

for some constant $\gamma > 0$. As all observations in the current sliding window can be assumed to be in steady-state when this condition is met, the truncation point is detected at point $l = i - N + 1$, at the beginning of the window. The window size N is included in the right-hand-side of the inequality as E_t itself is proportional to N, being the sum of N errors.

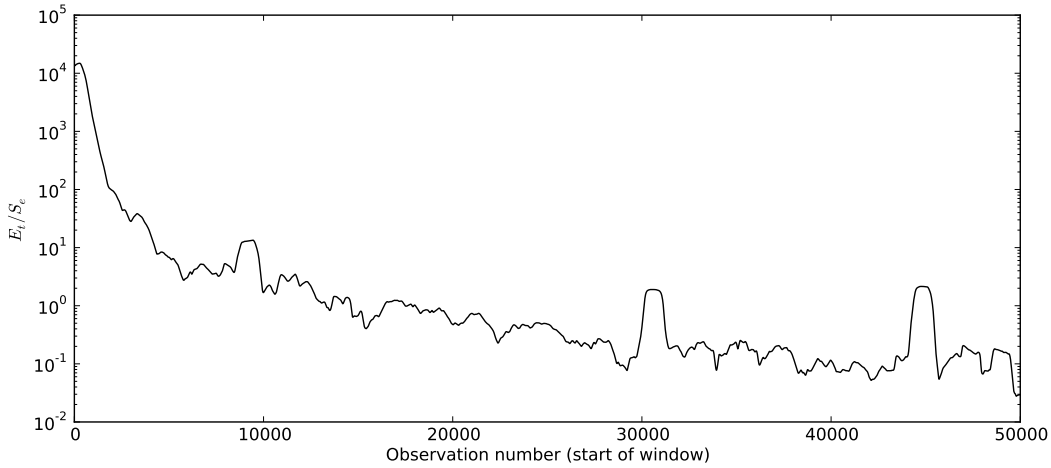


Figure 3.2: Time-series of the ratio E_t/S_e for the system states of an M/M/1 queue with $\rho = 0.9$ and initialised with 100 customers in the system. In this example, $N = 1,000$ and $\alpha = 0.01$.

One issue here is that the sample standard deviation S_e is calculated from data that includes the

initial transient phase, which will probably artificially inflate its value because of its non-stationarity and thus increase variance about C_t. Experimental testing showed that this effect could be mitigated by taking the statistic E_t to be the sum of squared forecasting errors rather than simply absolute errors. This new calculation for E_t is given by

$$E_t = \sum_{i=0}^{N-1} e_{t-i}^2, \qquad t \geq N-1. \tag{3.3}$$

The sum of squared errors given in Equation 3.3 increases relatively to the sum of absolute errors in Equation 3.3 as the sample standard deviation of the forecasting errors increases. Hence, this diminishes the effect of an inflated sample standard deviation due to the presence of an initial transient phase. Figure 3.2 shows an example of the behaviour of E_t relative to that of S_e across the output of a single run of an M/M/1 queue. The value of E_t/S_e generally decreases as the window moves through further observations, so choosing smaller values for γ should give longer truncation points.

The sequential algorithm for this method using forecasting methods to determine the horizontal flatness of the cumulative means plot is summarised in Figure 3.3.

Given window size N, smoothing factor α, and detection condition constant γ:

1. Obtain next observation X_t.
2. Calculate cumulative sum $c_t = c_{t-1} + X_t$ and update cumulative mean $C_t = \frac{c_t}{t+1}$.
3. Calculate new smoothed value $s_t = \alpha C_t + (1-\alpha)s_{t-1}$.
4. Compute forecasting error $e_t = s_t - C_t$.
5. If $t < N-1$, increment t by one and go to 1.
6. Calculate sum of squared errors, $E_t = \sum_{i=0}^{N-1} e_{t-i}^2$.
7. Compute sample standard deviation of the forecasting errors, $S_e = \sqrt{\frac{1}{t-1}\sum_{i=0}^{t-1}(e_t - \overline{e})^2}$.
8. If $E_t > \gamma N S_e$, increment i by one and go to 1.
9. Stop and return truncation point given by position $i - N + 1$.

Figure 3.3: Algorithm for the new forecasting steady-state detection method.

4 Calibration of Method Parameters

Values for the smoothing factor α, window size N and detection condition constant γ need to be calibrated to give a truncation rule that successfully detects the onset of steady-state for a range of simulation output.

4.1 Methodology

A truncation method accurately detecting steady-state should truncate at a point that falls within or sufficiently close to the region of steady-state behaviour of the simulated process. This means that the observation value at the point immediately following the truncation point—that is, the first observation of the truncated sample—should have a frequency distribution that approximately conforms to the true steady-state distribution of the process, as specified by Equation 2.1. To the extent that the distribution of values at this point do not match the expected steady-state distribution (beyond sampling error) for a given simulation model, the method has a systematic propensity to truncate at values that are not necessarily drawn from the process's steady-state, and this can cause bias in steady-state estimates calculated from the truncated sample. Thus, as one measure of a method's ability to detect the onset of steady-state, the empirical distribution observed at observations immediately following the truncation point can be compared to its steady-state distribution. This is only possible if this latter distribution can be calculated analytically or is otherwise known in advance.

This approach is used to determine which values for parameters α, N and γ are best suited to accurately detect steady-state on the output of a given simulation model. The average length of truncation can also be analysed to ensure that the combination of parameters do not give excessively long truncation points, that is, large truncation points without significant gain in conformance of the observations to their theoretical steady-state distributions.

Values for α (0.1 or 0.01), N (500, $1,000$ or $2,000$) and γ (0.1, 0.5, 1 or 2) are considered. These values for α and N were chosen based on preliminary testing, which looked for the range of values that would give credible results for a range of simulation processes. Lower values of α were not considered as these gave truncation points well in excess of the end of the initial transient phase in analytically tractable cases. The candidate values for γ were chosen by visually analysing plots of the ratio of E_t to S_e for a number of simulation models, and determining values for γ that may truncate

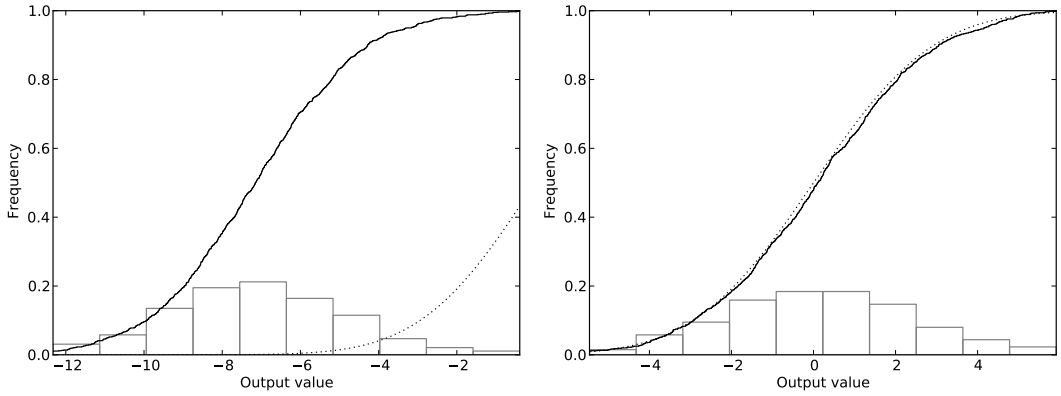
most or all of the initial transient phase when the condition given in Equation 3.2 is met. Figure 3.2 shows an example of such a plot.

For each combination of values for α, N and γ, the forecasting truncation method outlined in Chapter 3 is tested upon a range of simulation models which is based on the suite of models described by Eickhoff [5]. These models are:

- M/M/1 queueing models, with traffic intensities $\rho = 0.5$ or 0.9, using either response times or the number of customers in the system, and initialised either empty-and-idle or with 100 customers in the system.
- M/E_2/1 queueing models, with traffic intensities $\rho = 0.5$ or 0.9, using waiting times of customers in the queue, initialised either empty and idle or with 100 customers.
- M/H_2/1 queueing models, with traffic intensities $\rho = 0.5$ or 0.9, using waiting times of customers in the queue, initialised either empty and idle or with 100 customers, with the service time having a coefficient of variation $c_v = \sqrt{5}$.
- AR(1) autoregressive models, with parameters $\phi = -0.9$, 0, 0.9 or 0.99, and initial biases $b = 0$ or 100.
- Geometrical ARMA(k, k) (autoregressive–moving-average) models, of the orders $k = 1$ or 2, initialised with $X_{-1} = X_{-2} = 0$.
- Damped vibration process, with an amplitude of $k = 10$, a period of $T = 50$, and a transient length of $l = 250$ observations.
- Quadratic displacement process, with an initial displacement of 10 and a transient length $l = 100$ observations.
- Quadratic stretch process, with an amplitude of $k = 10$ and a transient length of $l = 100$.
- M/G/1 queueing models with Pareto service-time distributions using shape parameter $\alpha = 2.25$, and traffic intensities of $\rho = 0.8$, 0.9 or 0.95, observing waiting times, and initialised empty and idle.

The full specifications of these models are given in Appendix A.

An empirical distribution of the observations immediately following the truncation point for each combination of parameters and simulation models is obtained by running $1,000$ independent simulation runs. The observations given by each combination can be transformed into a *cumulative distribution function* (CDF) and, assuming that the steady-state distribution of these models can be calculated through theoretical analysis, this can be plotted alongside and visually compared to the corresponding theoretical steady-state CDF. The details for the computation of the steady-state distributions of the different models are given in Appendix A.



(a) $\alpha = 0.1, \bar{l} = 24$ (b) $\alpha = 0.01, \bar{l} = 281$

Figure 4.1: AR(1) model with parameter $\varphi = -0.9$ and initial bias $b = 100$. $N = 1,000$ and $\gamma = 0.1$. The theoretical steady-state CDFs are shown as dotted lines, the empirical CDFs are shown as solid lines, and the collected observations are shown by histograms in grey.

These simulation models and tests were programmed in C++. The WELL44497a pseudo-random number generator (PRNG) [28] was used to generate random numbers for the simulation runs, having good stochastic properties and guaranteeing extremely long cycles, [11]. The independent simulations were run serially, with each successive run using the state of the PRNG that resulted from the end of the previous run.

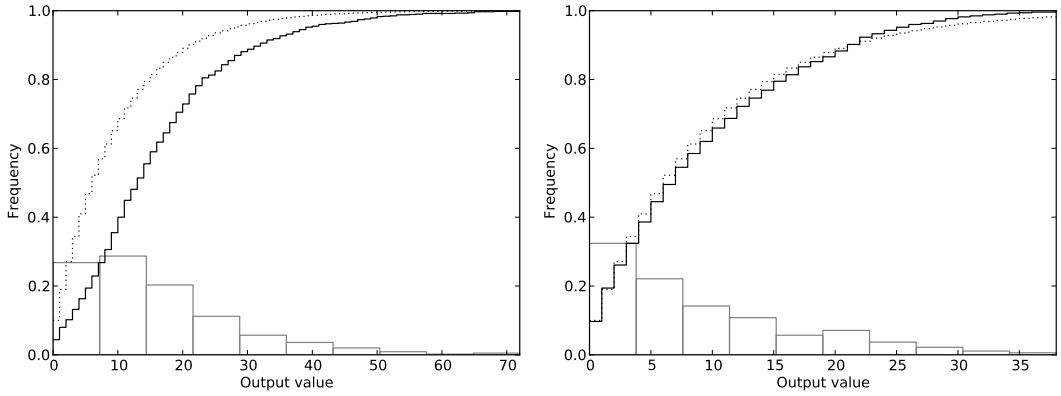
4.2 Results

All figures in this section show the CDF for the output observation values immediately following the truncation point as solid black lines, the expected steady-state CDF as dashed black lines, and the histogram of collected observations in grey. The average truncation points for a given model are specified by $\bar{l}$.

4.2.1 Smoothing Factor α

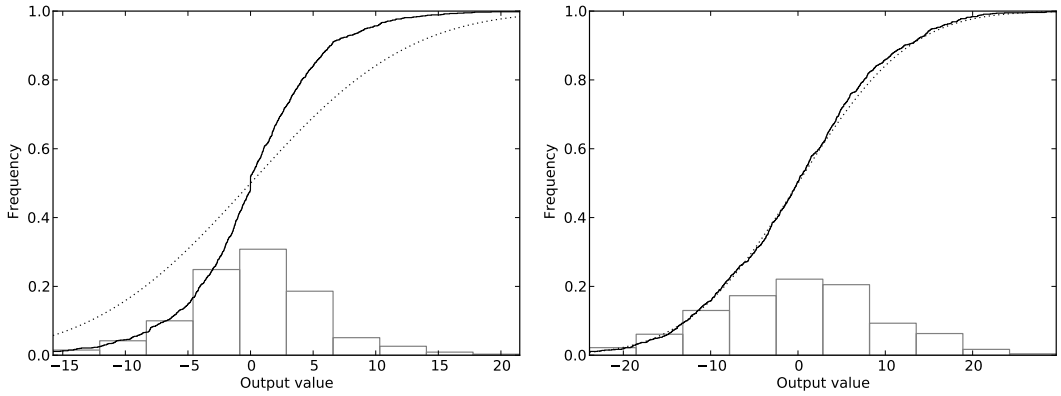
Lower values for the smoothing factor α will give greater smoothing of s_t. Consequently, at these lower values, it will take more observations for s_t to converge to the cumulative mean C_t and thus for values of e_t to decrease. This is because earlier values of C_t—when the values are more likely to be biased away from the true steady-state mean $E[X_\infty]$ which C_t later converges to—are more influential upon s_t in this case. Therefore, smaller α will increase the number of observations needed for the test statistic E_t to meet a given detection condition, as given by Equation 3.2, generally increasing the resulting truncation point l. The increased smoothing also makes the values of e_t less susceptible to potential outliers in the output.

For a number of the simulation models tested, varying α (0.1 or 0.01) had little effect on the accuracy of the distribution of observations following the truncation point in approximating the true



(a) $\alpha = 0.1, \bar{l} = 202$ (b) $\alpha = 0.01, \bar{l} = 1064$

Figure 4.2: System states of an M/M/1 queue, with traffic intensity $\rho = 0.9$ and initialised empty and idle. $N = 1,000$ and $\gamma = 0.1$. The theoretical steady-state CDFs are shown as dotted lines, the empirical CDFs are shown as solid lines, and the collected observations are shown by histograms in grey.



(a) $\alpha = 0.1, \bar{l} = 33$ (b) $\alpha = 0.01, \bar{l} = 187$

Figure 4.3: Quadratic stretch process with stretch factor $k = 10$ and transient length $l = 100$. $N = 1,000$ and $\gamma = 0.1$. The theoretical steady-state CDFs are shown as dotted lines, the empirical CDFs are shown as solid lines, and the collected observations are shown by histograms in grey.

steady-state distribution. This is presumably because the truncation points given by $\alpha = 0.1$ find the onset of steady-state to a sufficient degree, so later truncation points as generally given by $\alpha = 0.01$ provide negligible gain in accuracy in these cases.

A smoothing factor of $\alpha = 0.1$ did not give appropriate truncation points for some models that $\alpha = 0.01$ did. Figure 4.1 shows the distributions given by both values of α for an initially-biased AR(1) process. Using $\alpha = 0.1$ resulted in significant bias remaining after the truncation point, as shown by the empirical distribution clearly shifted to the left from the expected steady-state distribution in Figure 4.1a. This was not the issue for $\alpha = 0.01$, which very closely approximated the steady-state distribution, as Figure 4.1b shows. The inability to correctly truncate the initial transient is also shown

the average truncation points $\bar{l}$: $\alpha = 0.1$ gives $\bar{l} = 24$ and $\alpha = 0.01$ has $\bar{l} = 281$.

A similar effect occurs for a M/M/1 queue as shown in Figure 4.2, although here it is not as significant. The use of $\alpha = 0.01$ in Figure 4.2b is much better in approximating the steady-state distribution compared to $\alpha = 0.1$ in Figure 4.2a. The average truncation points again demonstrate the large value of α resulting in much smaller truncation points and thus not effectively mitigating initial bias, with $\alpha = 0.1$ giving $\bar{l} = 202$ and $\alpha = 0.01$ giving $\bar{l} = 1064$.

Again, Figure 4.3 demonstrates the lower α value not truncating initial bias suitably. This is applied to a quadratic stretch process, which does not have a non-stationary mean, but transient variance in the initial transient phase. $\alpha = 0.01$ is shown to approximate steady-state well in Figure 4.3b, unlike $\alpha = 0.1$ in Figure 4.3a, in which the distribution is not fully "stretched" to its steady-state.

Due to the failure of $\alpha = 0.1$ to effectively mitigate the initial transient in cases where $\alpha = 0.01$ is able to, the latter is chosen as the preferred value to use for this method.

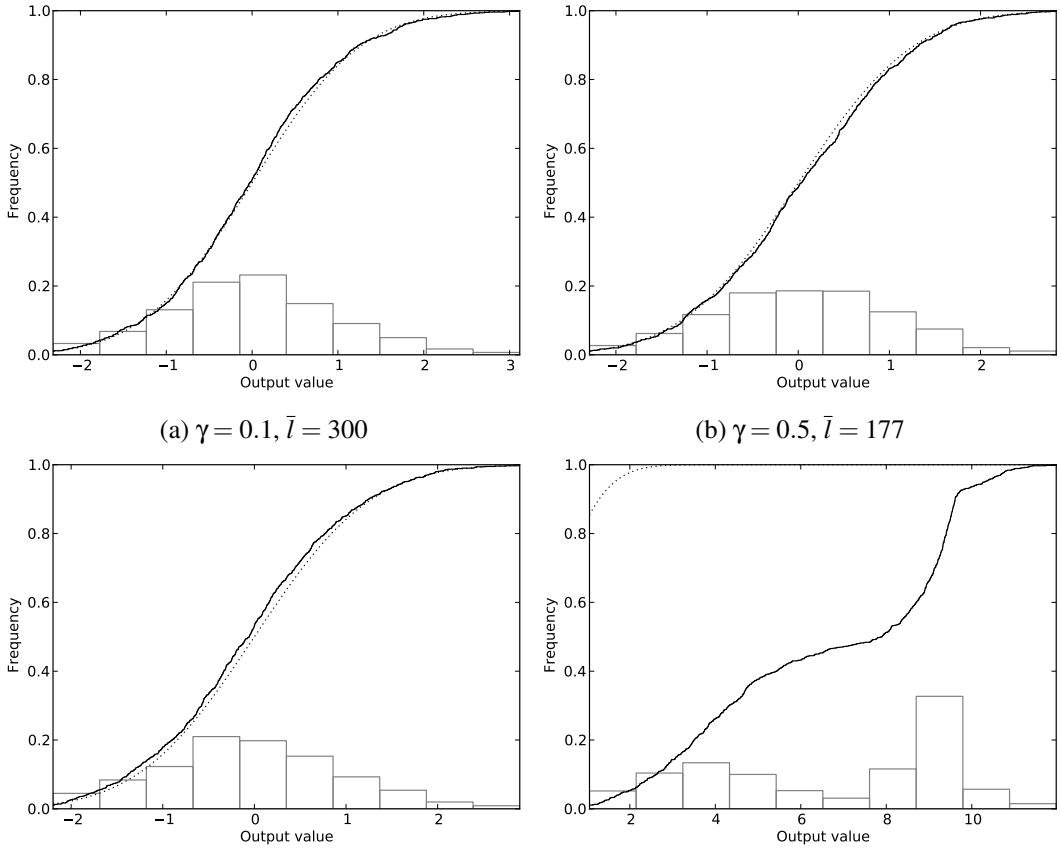
4.2.2 Detection Condition Constant γ

The detection condition constant γ simply specifies the relative magnitude of the detection criterion, as given by Equation 3.2. Larger values of γ will relax the condition by which a truncation point is found, yielding shorter truncation points; smaller values give a stricter condition and correspondingly longer truncation points. A value for γ needs to be determined that enables the method to truncate the initial transient phase effectively for a large range of simulation outputs, but that does not give excessively long truncation points, wasting observations and computation time.

Figure 4.4 gives the distributions resulting from the use of various values of γ (0.1, 0.5, 1 and 2) on the output from a quadratic displacement process, alongside the expected steady-state distribution for that process. Each of the values $\gamma = 0.1$, 0.5 and 1 show to approximate the steady-state distribution well, with negligible difference between each other, while $\gamma = 2$ in Figure 4.4d gives a strongly biased distribution, with almost all values above the steady-state mean $E[X_\infty] = 0$. The average truncation point $\bar{l}$ resulting from $\gamma = 0.1$ is $\bar{l} = 300$, which while noticeably bigger than that of $\gamma = 1$ at $\bar{l} = 121$, it is not a significant number of superfluous observations to truncate. The average point given when using $\gamma = 2$ lies well within the initial transient phase of the process, at $\bar{l} = 19$.

When applied to an $M/E_2/1$ queue with the results in Figure 4.5, the accuracy of the empirical distribution demonstrably decreases as γ is increased. Using $\gamma = 0.1$, as in Figure 4.5a, it shows to approximate steady-state reasonably well, with only a slightly greater concentration at low observations values compared to the steady-state distribution. For $\gamma = 2$, as in Figure 4.5d, there is a strong bias toward larger values, similar to that of the quadratic displacement model described above. However, unlike the quadratic displacement model, values of $\gamma = 0.5$ and 1 give intermediary levels of steady-state approximation accuracy. The smallest value of γ does not give excessively large truncation points either, with $\bar{l} = 2,014$, which is credible for a queue initialised with 100 customers in the system.

This same effect is shown for an $M/H_2/1$ queue in Figure 4.6; the accuracy of the observed distribution is increased at lower values of γ. The number of truncated data for $\gamma = 0.1$ is again not



(a) $\gamma = 0.1$, $\bar{l} = 300$

(b) $\gamma = 0.5$, $\bar{l} = 177$

(c) $\gamma = 1$, $\bar{l} = 121$

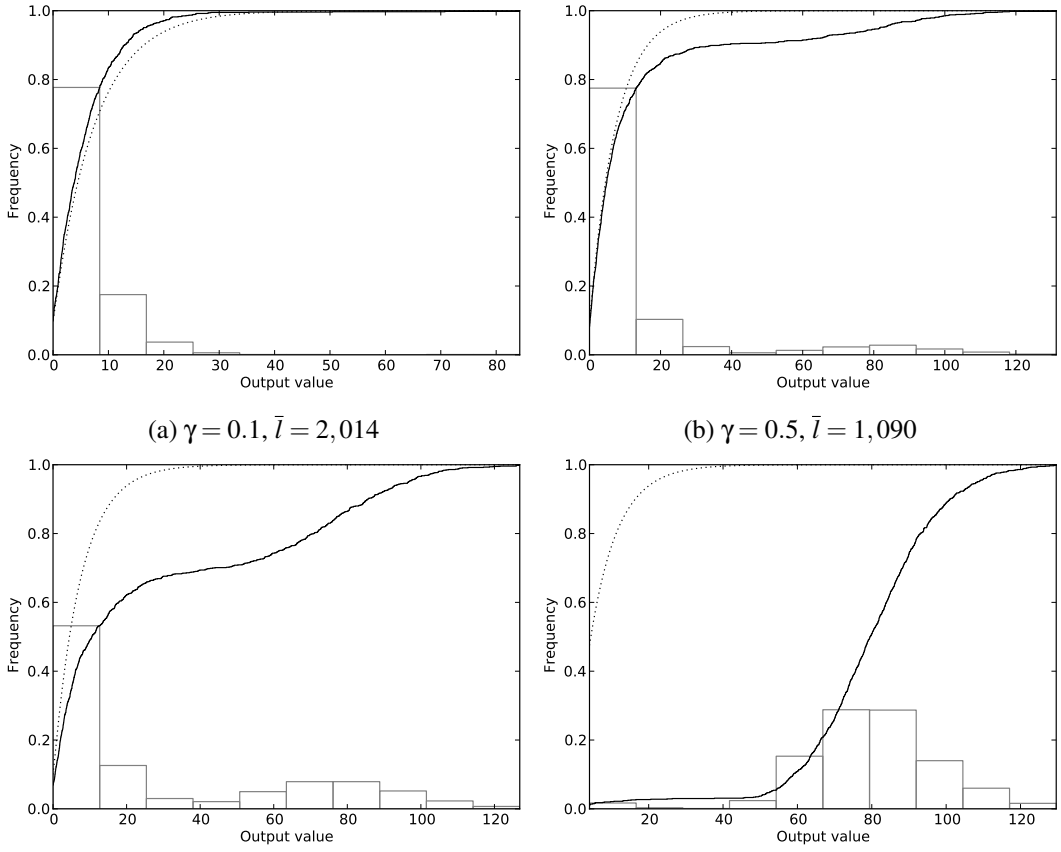
(d) $\gamma = 2$, $\bar{l} = 19$

Figure 4.4: Quadratic displacement process with displacement $k = 10$ and transient length $l = 100$. $N = 1,000$ and $\alpha = 0.01$. The theoretical steady-state CDFs are shown as dotted lines, the empirical CDFs are shown as solid lines, and the collected observations are shown by histograms in grey.

inordinate, at $\bar{l} = 2,014$ for the queue with 100 customers initially in the system. For all models tested, the lowest value $\gamma = 0.1$ proved to be superior is approximating the expected steady-state distribution, without giving truncation points of an impractical size. From the values that are tested, it is thus chosen as one of the most acceptable for use by the new truncation method.

4.2.3 Window Size N

Unlike both α and γ, neither increasing nor decreasing the window size N necessarily gives a more stringent or relaxed truncation criterion. This is because the truncation point is taken at the start of this sliding window, at point $t - N + 1$, so larger N will not necessarily return larger truncation points. Also, although the statistic E_t is the sum of N forecasting errors, the criterion it is compared against is also proportional to N (see Equation 3.2), so this will not inherently affect location of the truncation point in a systematic way either. However, the truncation points given by different window sizes N do affect the resulting truncation points due to the behaviour of E_t on N values for e_t as they converge to



(a) $\gamma = 0.1$, $\bar{l} = 2{,}014$

(b) $\gamma = 0.5$, $\bar{l} = 1{,}090$

(c) $\gamma = 1$, $\bar{l} = 708$

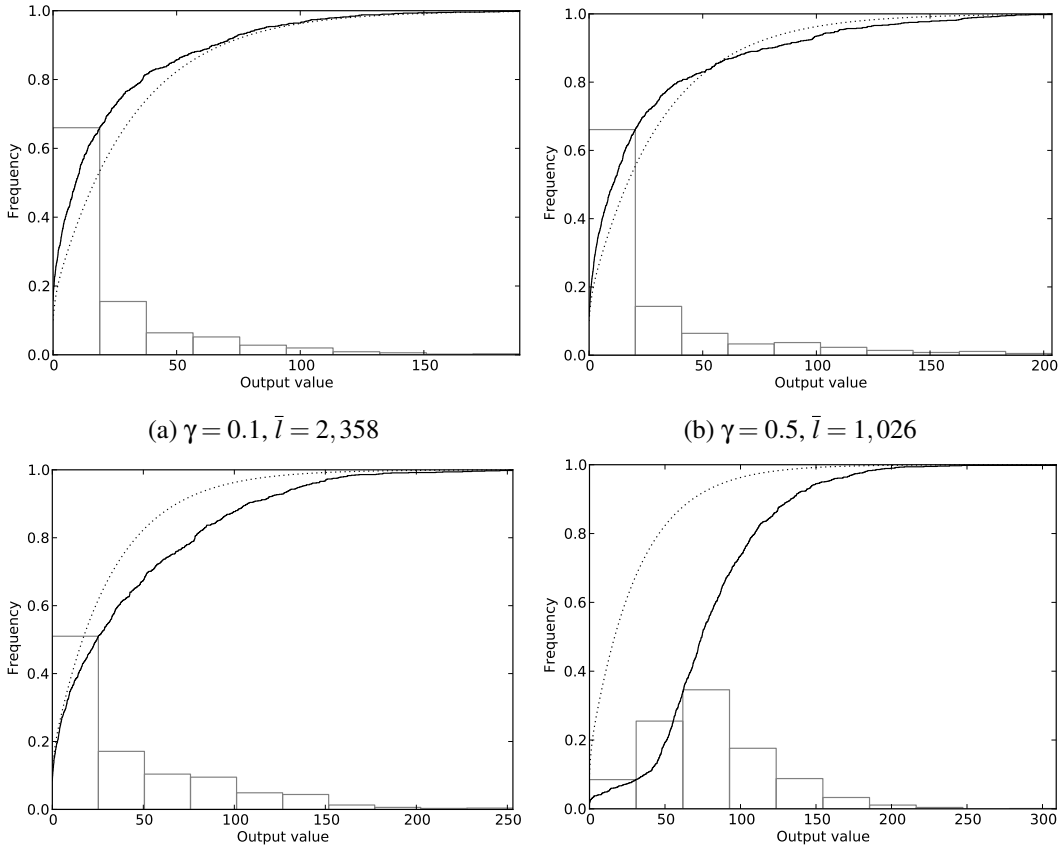
(d) $\gamma = 2$, $\bar{l} = 281$

Figure 4.5: Waiting times of an M/E_2/1 queue, with traffic intensity $\rho = 0.9$ and initialised with 100 customers in the system. $N = 1{,}000$ and $\alpha = 0.01$. The theoretical steady-state CDFs are shown as dotted lines, the empirical CDFs are shown as solid lines, and the collected observations are shown by histograms in grey.

zero.

The effect of varying N (500, 1,000 and 2,000) was negligible for many of the simulation models. However, a difference was noticeable for some models, for example the M/E_2/1 queue as shown in Figure 4.8. There was not much disparity in the average truncation points, with a minimum of $\bar{l} = 717$ for $N = 500$ and a maximum of $\bar{l} = 943$ for $N = 2{,}000$. Nevertheless, using $N = 1{,}000$, the method approximated the steady-state distribution exceedingly well, unlike for the other two values. With $N = 500$ in Figure 4.7a, the distribution was overly concentrated toward small values, while for $N = 2{,}000$ in Figure 4.7c, it was toward larger values. Of particular note is the fact that $N = 1{,}000$ gave a noticeably better distribution than that of $N = 2{,}000$, despite truncating at smaller points on average. The exact same effects as these are shown for an M/M/1 model in Figure 4.8.

Using values of $N = 1{,}000$ and $N = 2{,}000$ gave very similar results for many of the simulation models, including for an AR(1) process with autoregressive parameter $\varphi = 0.9$ and initial bias $b = 100$, waiting times of an empty-and-idle M/H_2/1 queue with traffic intensity $\rho = 0.5$, waiting times of an

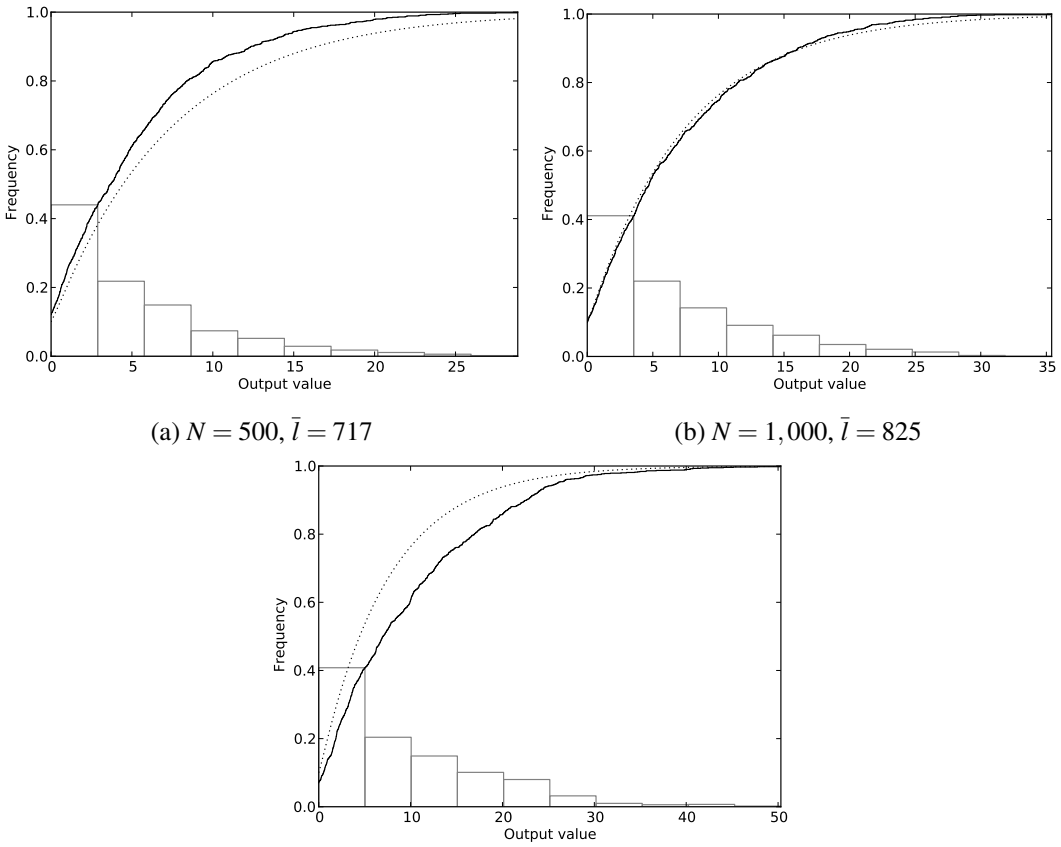


(a) $\gamma = 0.1$, $\bar{l} = 2,358$ (b) $\gamma = 0.5$, $\bar{l} = 1,026$

(c) $\gamma = 1$, $\bar{l} = 658$ (d) $\gamma = 2$, $\bar{l} = 335$

Figure 4.6: Waiting times of an M/H_2/1 queue, with traffic intensity $\rho = 0.9$ and initialised with 100 customers in the system. $N = 1,000$ and $\alpha = 0.01$. The theoretical steady-state CDFs are shown as dotted lines, the empirical CDFs are shown as solid lines, and the collected observations are shown by histograms in grey.

M/M/1 queue initialised with 100 customers in the system and $\rho = 0.5$, system states of an empty-and-idle M/M/1 queue with $\rho = 0.9$, and a quadratic displacement process with initial displacement $k = 10$ and initial transient length $l = 100$. Figure 4.9 displays an example of this for an initially-biased AR(1) process, where the resulting distributions from both values of N give near-perfect distributions to that of steady-state. In both cases, they also give very close values of the average truncation point, with $\bar{l} = 350$ for $N = 1,000$ and $\bar{l} = 321$ for $N = 2,000$. In these situations, nonetheless, using $N = 1,000$ is advantageous because it requires the look-ahead and analysis of fewer observations, thus it does not require the simulation model to generate as many observations and can compute a truncation point with less computation time. Hence, $N = 1,000$ is chosen as the window size for this method.

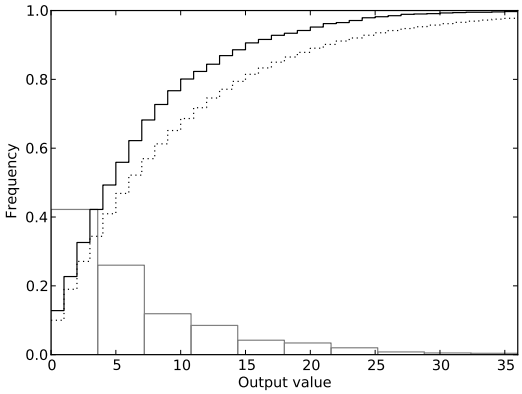


(a) $N = 500$, $\bar{l} = 717$

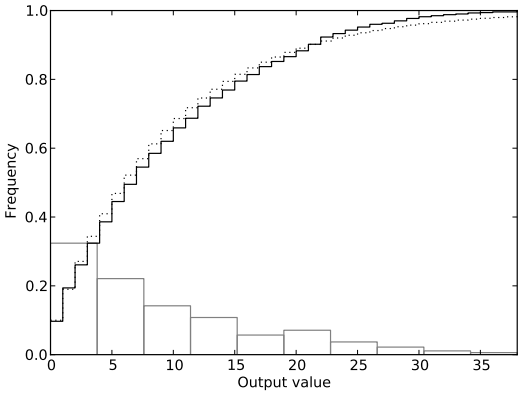
(b) $N = 1{,}000$, $\bar{l} = 825$

(c) $N = 2{,}000$, $\bar{l} = 943$

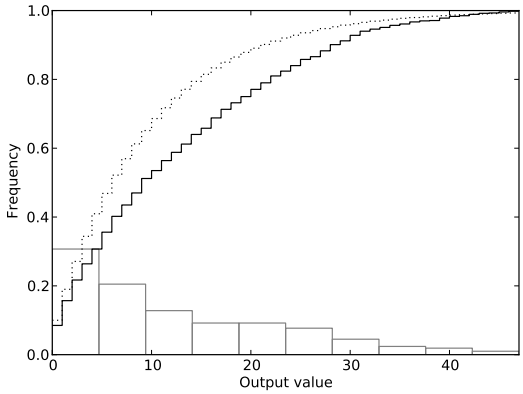
Figure 4.7: Waiting times of an M/E_2/1 queue, with traffic intensity $\rho = 0.9$ and initialised empty and idle. $\gamma = 0.1$ and $\alpha = 0.01$. The theoretical steady-state CDFs are shown as dotted lines, the empirical CDFs are shown as solid lines, and the collected observations are shown by histograms in grey.



(a) $N = 500$, $\bar{l} = 864$

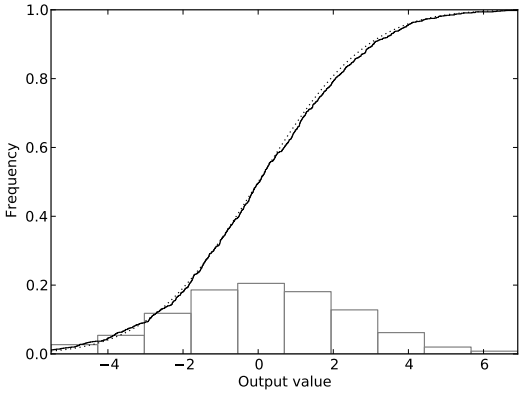


(b) $N = 1,000$, $\bar{l} = 1,064$

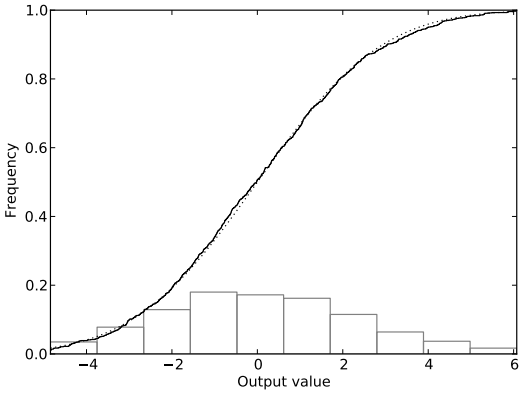


(c) $N = 2,000$, $\bar{l} = 1,229$

Figure 4.8: System states of an M/M/1 queue, with traffic intensity $\rho = 0.9$ and initialised empty and idle. $\gamma = 0.1$ and $\alpha = 0.01$. The theoretical steady-state CDFs are shown as dotted lines, the empirical CDFs are shown as solid lines, and the collected observations are shown by histograms in grey.



(a) $N = 1,000$, $\bar{l} = 350$



(b) $N = 2,000$, $\bar{l} = 321$

Figure 4.9: AR(1) process with parameter $\varphi = 0.9$ and initial bias $b = 100$. $\gamma = 0.1$ and $\alpha = 0.01$. The theoretical steady-state CDFs are shown as dotted lines, the empirical CDFs are shown as solid lines, and the collected observations are shown by histograms in grey.

4.3 Conclusions

A range of combinations of values for window size N (500, $1,000$ and $2,000$), smoothing factor α (0.01 and 0.1) and detection condition constant (0.1, 0.5, 1 and 2) were examined for calibrating the parameters of the forecasting method. The performance of each combination was evaluated by comparing the resulting distribution of the observations immediately following the truncation points to the steady-state distributions of those simulated processes, and this was done for a wide range of analytically tractable processes. On the basis of the study, it was found that values if $N = 1,000$, $\alpha = 0.01$ and $\gamma = 0.1$ were most suitable for truncating initialisation bias for a range of processes, and these values are used for further evaluation of our forecasting method in the following chapters.

5 Performance Evaluation of the New Method

Using the parameters obtained from Chapter 4—smoothing factor $\alpha = 0.01$, detection condition constant $\gamma = 0.1$ and window size $N = 1,000$—the developed forecasting truncation rule is compared to the sequential MSER-5 method (see Section 2.3.1), one of the most popularly advocated methods in the recent literature. The two methods are compared based on their ability to consistently determine a truncation point that falls within or sufficiently close to steady-state, namely, when the distribution of observations immediately following the truncation points can approximate the process's theoretical steady-state distribution.

5.1 Methodology

Default parameters of batch size $m = 5$ and a sequential increase amount of $z = 10\%$ were used for the sequential MSER-5 method. Since we are only concerned about sequential truncation methods in the context of single runs, the number of simultaneous replications $k = 1$. An initial run length of $n = 1,000$ was used as this has shown to give enhanced performance over the minimum value of $n = 100$ suggested by Hoad *et al.* [17], [10].

These two sequential truncation methods were applied to all the models and corresponding parameters as outlined in Section 4.1, with the exception of the M/G/1 queueing model with Pareto service-time distribution, as the sequential MSER-5 rule would not find truncation points in reasonable time for this model. Further details of the simulation models used are given in Appendix A.

As in Section 4.1, the empirical distributions of the output values immediately following each truncation point are given by the truncation methods are obtained from $1,000$ independent replications for each simulation model. The CDFs obtained from these are then plotted alongside the true steady-state distribution for each model, and visually compared against one another to determine which method is superior in finding truncation points that lie in steady-state. The average truncation points given by each method for each model are also compared.

Both our forecasting method and the MSER-5 method are performed on *paired* simulation runs. This means that, for each of the $1,000$ runs tested upon by one method on one simulation model, the other method is tested using these exact same runs for that given model. Specifically, the paired runs will have the entire state of the system initialised identically (for both the simulation model and

the PRNG). Of course, both these methods are sequential so they do not end up analysing the same data set for each of the runs, as one method will presumably require more observations than the other. However, the set of observations obtained from the shortest of both runs will correspond to the initial observations used in the longest run. Paired simulations runs thus ensure that the randomness inherent in the simulation output will not cause disparities in the results obtained using each truncation method.

The simulation tests were programmed in C++, and the simulation model implementations are the same as used in Chapter 4. The WELL44497a PRNG [28] is employed again, and the initial state of the PRNG used by each sequential paired simulation run is taken from the final state of the PRNG after the longest simulation run from the previous paired runs.

5.2 Results

The figures listed in this section and in Appendix B display the CDFs resulting from the new forecasting method as solid lines, from the sequential MSER-5 method as dashed lines, and the steady-state CDFs for the corresponding models are given as dotted lines. The average truncation method of our forecasting method is given by $\bar{l}_F$, and that of the MSER-5 method by $\bar{l}_M$.

The new forecasting method performed better than the sequential MSER-5 method for most models, and there was only one instance of the MSER-5 method noticeably outperforming the forecasting method—on the output of an AR(1) process with $\varphi = 0$ (see Figure 5.5). An example of our forecasting method clearly attaining a better approximation of the true steady-state distribution is given in Figure 5.1, an M/E_2/1 queue initialised with 100 customers in the system. Here, the distribution of the observation value immediately following the truncation point given by the forecasting method almost perfectly matches the expected steady-state distribution, while the MSER-5 method gives a distribution that heavily leans toward higher values. 50% of the values given by the steady-state distribution are equal to zero, which is matched by 50.1% given by the forecasting method, but only 1% of the observed values by the MSER-5 method are equal to zero. The MSER-5 method must have systematically found truncation points still within the initial transient phase, as given by its average truncation point $\bar{l}_M = 275$, unlike that of the forecasting method which returned significantly larger points on average, $\bar{l}_F = 846$, as detailed in Table 5.1.

Figure 5.2 shows another example of the forecasting method noticeably approximating the steady-state distribution more accurately than the MSER-5 method. The forecasting method again successfully finds a close approximation to the theoretical steady-state distribution. The MSER-5 method, however, in contrast to Figure 5.1, finds an excessively large number of output values of zero. This is because the MSER-5 method truncated at the first observation (such that $l = 0$) in most of these cases, which is a common artifact of the MSER-5 method for processes with small variance in their initial transient phase, [10]. This effect is seen with further exaggeration in Figure 5.3, for a quadratic stretch process. In this case, the MSER-5 method found a truncation point $l = 0$ for each of the $1,000$ simulation runs. The forecasting method gave relatively small truncation points, $\bar{l}_F = 194$, but still approximated the steady-state distribution accurately.

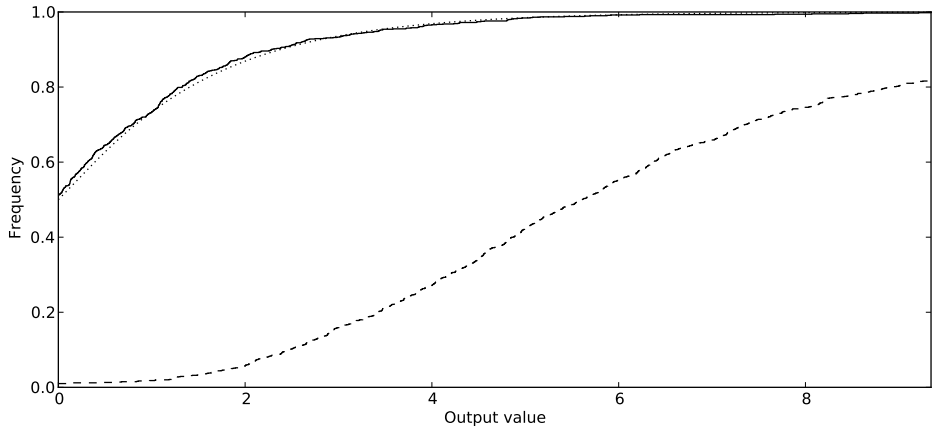


Figure 5.1: Waiting times of an M/E_2/1 queue, with traffic intensity $\rho = 0.5$ and 100 customers initially in the system. $l_F = 846$ and $l_M = 286$. The theoretical steady-state CDFs are shown as dotted lines, the empirical CDFs are shown as solid lines, and the collected observations are shown by histograms in grey.

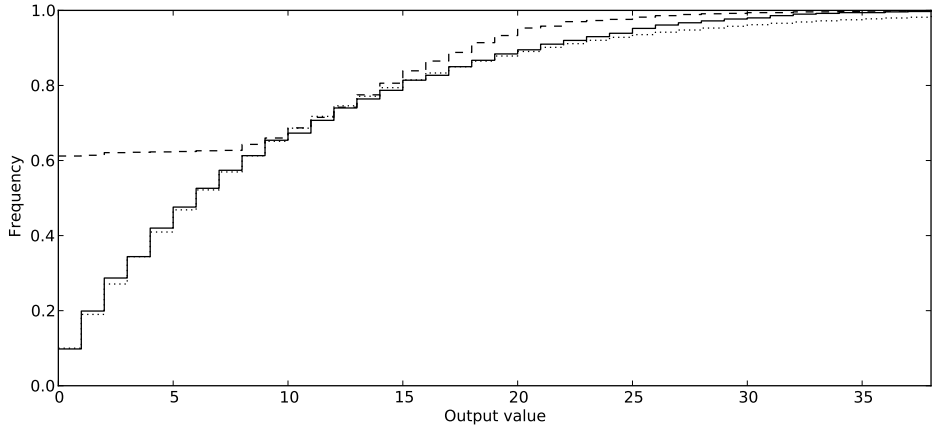


Figure 5.2: System states of an M/M/1 queue, with traffic intensity $\rho = 0.9$ and initialised empty and idle. $l_F = 1,029$ and $l_M = 195$. The theoretical steady-state CDFs are shown as dotted lines, the empirical CDFs are shown as solid lines, and the collected observations are shown by histograms in grey.

The case where both methods deviate most significantly from the true steady-state distribution is an AR(1) process with a negative parameter $\varphi = -0.9$, displayed in Figure 5.4. Both processes give greater numbers of observations near zero than the steady-state distribution. However, this effect is not drastic for either process, and the observations given by both give means that are close to the steady-state mean $E[X_\infty] = 0$, with the mean given by the forecasting method $X_{l_F} = -0.010$ and by the MSER-5 method $X_{l_F} = 0.006$.

The only instance in which the MSER-5 method outperformed the forecasting method is shown in Figure 5.5, and even here the performance of the MSER-5 method is considerably superior to that of the forecasting method, with both distributions varying from the steady-state by small amounts in

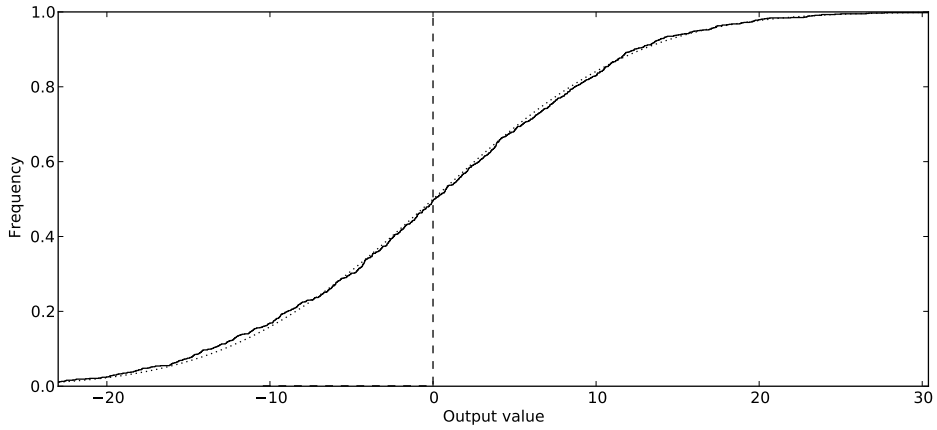


Figure 5.3: Quadratic stretch process with stretch factor $k = 10$ and transient length $l = 100$. $l_F = 187$ and $l_M = 0$. The theoretical steady-state CDFs are shown as dotted lines, the empirical CDFs are shown as solid lines, and the collected observations are shown by histograms in grey.

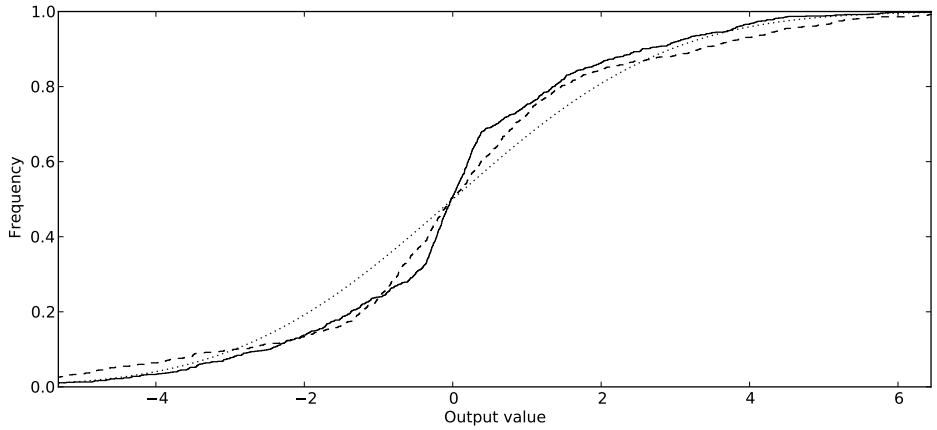


Figure 5.4: AR(1) process with autoregressive parameter $\varphi = -0.9$ and initial bias $b = 0$. $l_F = 34$ and $l_M = 10$. The theoretical steady-state CDFs are shown as dotted lines, the empirical CDFs are shown as solid lines, and the collected observations are shown by histograms in grey.

both the negative and positive directions. This was for an AR(1) process with parameter $\varphi = 0$, hence there is no correlation between observations and it is equivalent to a Gaussian white noise process. This also means that the initial bias b has no influence whatsoever on the output, so the cases where $b = 0$ and $b = 100$ are equivalent. The accuracy of the MSER-5 method here is achieved by the fact that it finds a truncation point $l = 0$ the majority of the time, and since this sequence is time-stationary and not autocorrelated, systematic selection of observations at a given point will give the steady-state distribution. The forecasting method generally truncated at later points, which is shown in Table 5.1 with an average $\bar{l}_F = 35$. The resulting distribution is overly concentrated around zero, as seen in Figure 5.5, so its determination of the truncation point tends to be at observations with values close to zero. The cumulative mean C_t is also probably very close to zero at this point, as there is no initial

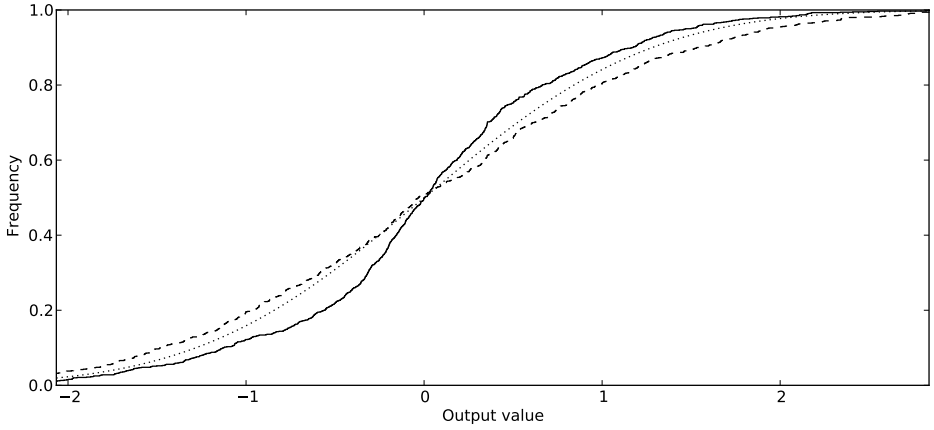


Figure 5.5: AR(1) process with autoregressive parameter $\varphi = 0$ and initial bias $b = 0$. $l_F = 38$ and $l_M = 5$. The theoretical steady-state CDFs are shown as dotted lines, the empirical CDFs are shown as solid lines, and the collected observations are shown by histograms in grey.

transient phase and the data are uncorrelated.

The figures for each model used in this experiment that are not shown in this section can be found in Appendix B. Table 5.1 outlines the average truncation points given by both methods for the models tested. The forecasting method consistently gave larger average truncation points than the MSER-5 method, with the exception of just the M/E_2/1 method using waiting times, traffic intensity $\rho = 0.5$, and initialised empty and idle. For a number of models, the MSER-5 method is seen to give very small truncation points, for example the ARMA(1,1) and quadratic stretch processes. This is a result of it returning truncation points at the first observation for a significant proportion of runs, determining no transient phase when one is actually present. No such behaviour was discovered for the forecasting method.

Table 5.1: Average truncation points given by the forecasting truncation method and the sequential MSER-5 method for each of the models. The average truncation point returned by the forecasting method for a given is denoted by $\bar{l}_F$, with a standard deviation by S_F. The average point for the MSER-5 rule is given by $\bar{l}_M$, with a standard deviation S_M.

Model	$\bar{\mathbf{l}}_\mathbf{F}$	$\mathbf{S}_\mathbf{F}$	$\bar{\mathbf{l}}_\mathbf{M}$	$\mathbf{S}_\mathbf{M}$
AR(1), $\varphi = 0.9$, $b = 0$	150.14	90.23	14.14	38.12
AR(1), $\varphi = 0.9$, $b = 100$	350.39	7.67	45.83	44.16
AR(1), $\varphi = 0.99$, $b = 0$	1,193.71	516.98	150.28	242.88
AR(1), $\varphi = 0.99$, $b = 100$	952.71	177.98	346.18	195.70
AR(1), $\varphi = 0$, $b = 0$	37.64	33.15	5.42	12.59
AR(1), $\varphi = 0$, $b = 100$	36.72	32.09	5.39	12.37
AR(1), $\varphi = -0.9$, $b = 0$	33.94	30.37	10.38	33.15
AR(1), $\varphi = -0.9$, $b = 100$	281.14	0.75	38.12	28.60
ARMA(1,1), $X_{-1} = 0$	66.88	43.66	5.69	14.76
ARMA(2,2), $X_{-2} = X_{-1} = 0$	132.72	45.11	8.91	22.90
Damped vibration, $k = 10$, $T = 50$, $l = 250$	182.94	5.24	218.72	34.39
M/E_2/1, waiting times, $\rho = 0.5$, 100 customers	846.11	58.60	268.46	135.25
M/E_2/1, waiting times, $\rho = 0.5$, empty and idle	68.64	48.09	91.87	171.29
M/E_2/1, waiting times, $\rho = 0.9$, 100 customers	2,014.25	541.44	926.74	465.67
M/E_2/1, waiting times, $\rho = 0.9$, empty and idle	877.58	595.53	206.46	316.51
M/H_2/1, waiting times, $\rho = 0.5$, 100 customers	841.48	136.70	319.40	204.876
M/H_2/1, waiting times, $\rho = 0.5$, empty and idle	445.69	346.42	163.90	245.82
M/H_2/1, waiting times, $\rho = 0.9$, 100 customers	2,359.97	1,200.32	670.88	701.47
M/H_2/1, waiting times, $\rho = 0.9$, empty and idle	2,629.63	1,620.67	298.78	503.78
M/M/1, system states, $\rho = 0.5$, 100 customers	604.09	84.35	163.58	138.23
M/M/1, system states, $\rho = 0.5$, empty and idle	87.90	52.74	81.48	157.26
M/M/1, response times, $\rho = 0.5$, 100 customers	842.16	71.40	251.50	111.88
M/M/1, response times, $\rho = 0.5$, empty and idle	107.95	80.64	84.67	161.15
M/M/1, system states, $\rho = 0.9$, 100 customers	2,822.20	1,076.02	783.29	459.37
M/M/1, system states, $\rho = 0.9$, empty and idle	1,029.17	709.63	194.86	319.97
M/M/1, response times, $\rho = 0.9$, 100 customers	1,998.94	639.51	834.34	449.01
M/M/1, response times, $\rho = 0.9$, empty and idle	1,088.64	708.67	209.40	319.61
Quadratic displacement, $k = 10$, $l = 100$	299.75	5.09	75.63	14.47
Quadratic stretch, $k = 10$, $l = 100$	187.80	92.44	0.11	3.48
Random walk	3,585.68	2,001.29	1,001.12	2,006.13

6 Discussion

6.1 Findings

Values for smoothing factor $\alpha = 0.01$, detection condition constant $\gamma = 0.1$ and window size $N = 1,000$ were found that allow the new method to accurately determine the onset of steady-state for a range of simulation processes, as determined by the closeness of the distributions of observation values immediately following the truncation point to their expected steady-state distributions. While these values gave highly accurate steady-state distributions for most of the simulation models, there were some for which these values gave distributions that only roughly approximated steady-state. This is because the convergence of the mean to its steady-state value does not guarantee the convergence of the entire distribution to steady-state, although this can sometimes be the case. Nevertheless, when stricter conditions on the convergence of the mean to steady-state are in place—such as the requirement of a flatter and more horizontal time-series of the cumulative mean—the entire distribution should also more accurately represent the steady-state distribution. As we are only looking at the observation immediately following the truncation point, it is not crucial for the resulting distribution to conform to the steady-state one to an exceptional degree. The reason for this is that final point estimates are calculated from the entire truncated sample, that is, from all observations from the truncation point onwards. Thus, it is sufficient for the truncation point to be at a point approaching steady-state rather than within steady-state, and any systematic bias will be avoided if the mean value at the truncation point is close to the steady-state mean.

Large truncation points mean that relatively many observations must be generated before the estimation of steady-state measures can even begin. Deletion of the initial transient phase is necessary in cases where long initial transient phases do occur in the output, to ensure unbiased estimators. For typical simulations, however, the initial transient phase should comprise only a small fraction of the total number of output values that need to be collected, so the deletion of this will generally incur negligible computing resources and time.

The sequential version of the MSER-5 method has been advocated as an effective technique for automated detection and deletion of the initial transient phase, [17]. However, in terms of determining truncation points that lie within or near steady-state, it is shown to be inaccurate for a number of output processes. In almost all cases tested, the new method proposed in this report determined points that gave distributions that approximated steady-state more accurately than the MSER-5 method. The MSER-5 method failed to even give accurate steady-state means for the observations following the

truncation point. This is largely because of its tendency to find inordinate numbers of truncation points at the first observation in the output of processes with small variance in their initial transient phase, which is common in stable queueing models initialised empty and idle. Bias will occur in estimates calculated from the truncated sample due to this systematic tendency of the truncation point to occur at values unrepresentative of steady-state. For example, empty and idle queueing systems will have their first observations collected from customers who arrive into an empty system, which will generally be values lower than the steady-state mean of the system.

Unlike the truncation methods that have been proposed in the literature to date, the new forecasting method presented in this report shows to be promising for use in sequential simulation analysis. This is partly attributed to the fact that it is an inherently sequential method, unlike many other sequential methods that are adapted to the sequential context from non-sequential simulation. This means that the new method looks ahead until it determines the onset of steady-state while sequentially collecting further observations, whereas methods based in non-sequential techniques—such as MSER-5—look for the best truncation point in a given sample before potentially collecting further data. This means that they have a tendency to find earlier truncation points to reduce the variance of truncated sample estimates, which may be necessary in fixed-sample-size truncation but not in sequential analysis. This new forecasting method thus shows to be a good candidate for use as an automated transient deletion method in *Akaroa2* as well as other commercial simulation packages. An implementation for *Akaroa2* is given in Appendix C, using an online sequential algorithm for calculating S_e, [37].

6.2 Future Work

Although the method is grounded in the convergence of the cumulative mean to its steady-state value, which is guaranteed for any stable simulation, its technique of determining the flatness of the cumulative mean time-series does not necessarily give assurance about the level of convergence. This is dependent upon the ratio of the sum of squared forecasting errors in the sliding window E_t to the sample standard deviation of all forecasting errors S_e, as given by Equations 3.2 and 3.3. While it is sure that the value of E_t is significantly larger during the initial transient phase and converging to zero during steady-state, and S_e also converges to zero though much more slowly, the specific behaviour of the convergence of S_e is not necessarily known. Different types of non-stationarity in the initial transient phase of simulation models could may adversely affect this behaviour. For all of the models tested in this report, however, the detection condition gave adequate truncation points. Deriving a theoretical justification for the method's effectiveness in determining truncation points would help to ensure its universal applicability.

The calculation of the sample standard deviation S_e is based upon data derived from observations that include an initial transient phase. Methods for calculating S_e while minimising any effect the initial transient phase has on e_t could help to increase the accuracy of the method, although the exact magnitude of this effect on the current calculation of S_e is unknown. Future investigations into alternative methods for calculating this could suggest a more appropriate method.

Adequate values for the parameter values of α, γ and N were found, as in Chapter 4, but there were only chosen from a limited number of possible values that were analysed. It is known that the values for α and γ can be reduced to give stricter—hence longer—truncation points, so the effect of varying these values can be extrapolated. (Smaller values are potentially unideal because of possible overestimation of truncation points and thus wasted computation resources.) Unlike these, the behaviour of varying N is not predictable, so it would be beneficial to look into a greater range of possible values for this parameter. Large values for N may give more confidence that the process has in fact converged close to steady-state. Although, very large values for N could give wasted observations when this number exceeds the number of steady-state observations that are analysed.

The method's efficacy was evaluated over a variety of simulation models; a similar collection to that used in Eickhoff, [5]. However, this is certainly not exhaustive of all types of non-stationarity in the simulation output. For example, there was no model tested on with pronounced negative initialisation bias that slowly converges to steady-state. Different forms of variance, autocorrelations and oscillations, for instance, could also occur in real-world simulation output. The range of models used in this research was limited by both time and the need for *a priori* knowledge of the steady-state characteristics of the models to evaluate the truncation method. For complex simulation systems that would be used in practice, these steady-state values may be impossible to know in advance. Thus, future research into this method for further types of simulation models would help to establish it as a universal method.

7 Conclusions

Many methods for truncating the initial transient phase have been proposed in the literature, but none have shown to effectively locate the onset of steady-state in the context of single-run sequential simulation. This report proposes a new automated sequential truncation method, which is based on the convergence of the cumulative mean to its steady-state value, and uses forecasting techniques to determine this convergence. Suitable parameters for this method are found to enable it to detect the onset of steady-state effectively for a range of simulation models. Its performance is found to be superior than a sequential version of the popularly advocated MSER-5 method across most simulation models. This new method consistently found truncation points that lie close to or within the steady-state phase for a range of models, without giving excessively large truncation points and potentially wasting computational time. This method thus appears to be a robust automated sequential truncation method and a good candidate for implementation in *Akaroa2* and other commercial simulation packages that utilise sequential analysis. An *Akaroa2* implementation of the method is also developed.

Further research across an even wider range of simulation models would help to confidently establish this method as an all-purpose truncation method and potentially suggest refinements to the method's parameters. A very attractive feature of the new method is that it is conceptually simple, as it is based on the convergence of the cumulative mean to its steady-state value. Since such convergence to steady-state is typical for any parameter of the probability distribution of states of a stable stochastic system, it could be easily adapted in sequential analysis of variances, quantiles, and other such measures.

Bibliography

[1] Joseph Abate and Ward Whitt. Transient behavior of regulated brownian motion, II: Non-zero initial conditions. *Advances in Applied Probability*, 19(3):pp. 599–631, 1987.

[2] Philip G. Brabazon. Using slithers of simulation in a new approach for intelligent initialization of non-terminating systems. In *Proceedings of the 40th Conference on Winter Simulation*, WSC '08, pages 547–555. Winter Simulation Conference, 2008.

[3] Robert G. Brown, Richard F. Meyer, and D. A. D'Esopo. The fundamental theorem of exponential smoothing. *Operations Research*, 9(5):pp. 673–687, 1961.

[4] Richard W. Conway. Some tactical problems in digital simulation. *Management Science*, 10(1): pp. 47–61, 1963.

[5] Mirko Eickhoff. *Sequential Analysis of Quantiles and Probability Distributions by Replicated Simulations*. PhD thesis, Department of Computer Science and Software Engineering, University of Canterbury, Christchurch, New Zealand, 2007.

[6] Mirko Eickhoff, Don McNickle, and Krzysztof Pawlikowski. Detecting the duration of initial transient in steady state simulation of arbitrary performance measures. In *Proceedings of the 2nd International Conference on Performance Evaluation Methodologies and Tools*, ValueTools '07, pages 42:1–42:7, ICST, Brussels, Belgium, Belgium, 2007. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[7] Gregory Ewing, Don McNickle, and Krzysztof Pawlikowski. Credibility of the final results from quantitative stochastic simulation. In *EUROSIM'95*, pages 189–194, 1995.

[8] Gregory C. Ewing, Krzysztof Pawlikowski, and Don McNickle. Akaroa2: Exploiting network computing by distributing stochastic simulation. In *International Society for Computer Simulation*, pages 175–181, 1999.

[9] William W. Franklin and K. Preston White, Jr. Stationarity tests and MSER-5: exploring the intuition behind mean-squared-error-reduction in detecting and correcting initialization bias. In *Proceedings of the 40th Conference on Winter Simulation*, WSC '08, pages 541–546. Winter Simulation Conference, 2008.

[10] Adam Freeth, Krzysztof Pawlikowski, and Don McNickle. Searching for effective methods for detecting the onset of steady-state in quantitative discrete-event simulation. In preparation.

[11] Adam Freeth, Krzysztof Pawlikowski, and Don McNickle. Pseudo-random number generators for massively parallel discrete-event simulation. Technical Report TR-COSC 01/12, University of Canterbury, Christchurch, New Zealand, 2012.

[12] Everette S. Gardner. Exponential smoothing: The state of the art. *Journal of Forecasting*, 4(1): 1–28, 1985.

[13] Behshid Ghorbani. Initial transient phase of steady state simulation: methods of its length detection and their evaluation in Akaroa2. Master's thesis, Department of Computer Science, University of Canterbury, 2004.

[14] Geoffrey Gordon. *System Simulation*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition, 1977.

[15] Winfried Grassmann. Rethinking the initialization bias problem in steady-state discrete event simulation. In *Proceedings of the 2011 Winter Simulation Conference*, pages 593–599, Piscataway, NJ, 2011. IEEE.

[16] K. Hoad and S. Robinson. Implementing MSER-5 in commercial simulation software and its wider implications. In *Proceedings of the 2011 Winter Simulation Conference*, pages 495–503, Piscataway, NJ, 2011. IEEE.

[17] Kathryn Hoad, Stewart Robinson, and Ruth Davies. Automating discrete event simulation output analysis – automatic estimation of number of replications, warm-up period and run length. In *INFORMS Simulation Society Research Workshop*. INFORMS Simulation Society, Warwick, Coventry, 2009.

[18] Kathryn Hoad, Stewart Robinson, and Ruth Davies. Automating warm-up length estimation. *Journal of the Operational Research Society*, 61(9):1389–1403, 2010.

[19] W. David Kelton. Transient exponential-Erlang queues and steady-state simulation. *Commun. ACM*, 28(7):741–749, July 1985.

[20] Gerald T. Mackulak, Sungmin Park, John W. Fowler, Sonia E. Leach, and J. Bert Keats. A sequential stopping rule for a steady-state simulation based on time-series forecasting. *Simulation*, 78(11):643–654, 2002.

[21] Prasad S. Mahajan and Ricki G. Ingalls. Evaluation of methods used to detect warm-up period in steady state simulation. In *Proceedings of the 36th Conference on Winter Simulation*, WSC '04, pages 663–671. Winter Simulation Conference, 2004.

[22] Mary A. McClarnon. Detection of steady state in discrete event dynamic systems: an analysis of heuristics. Master's thesis, School of Engineering and Applied Science, University of Virginia, 1990.

[23] Don McNickle. Maple program for calculating M/E_2/1 waiting times. Personal communication, November 2010.

[24] Don McNickle. Maple program for calculating M/H_2/1 waiting times. Personal communication, November 2010.

[25] Don McNickle, Gregory C. Ewing, and Krzysztof Pawlikowski. Some effects of transient deletion on sequential steady-state simulation. *Simulation Modelling Practice and Theory*, 18(2): 177–189, 2010.

[26] Don McNickle, Krzysztof Pawlikowski, and Greg Ewing. Akaroa2: A controller of discrete-event simulation which exploits the distributed computing resources of networks. In *24th European Conference on Modelling and Simulation*, 2010.

[27] Anup C. Mokashi, Jeremy J. Tejada, Saeideh Yousefi, Ali Tafazzoli, Tianxiang Xu, James R. Wilson, and Natalie M. Steiger. Performance comparison of MSER-5 and N-Skart on the simulation start-up problem. In *Winter Simulation Conference'10*, pages 971–982, 2010.

[28] François Panneton, Pierre L'Ecuyer, and Makoto Matsumoto. Improved long-period generators based on linear recurrences modulo 2. *ACM Trans. Math. Softw.*, 32(1):1–16, March 2006.

[29] Raghu Pasupathy and Bruce Schmeiser. The initial transient in steady-state point estimation: contexts, a bibliography, the mse criterion, and the MSER statistic. In *Proceedings of the 42nd Conference on Winter Simulation*, WSC '10, pages 184–197. Winter Simulation Conference, 2010.

[30] Krzysztof Pawlikowski. Steady-state simulation of queueing processes: survey of problems and solutions. *ACM Computing Surveys*, 22:123–170, June 1990.

[31] Krzysztof Pawlikowski. Towards credible and fast quantitative stochastic simulation. In *Proceedings of International SCS Conference on Design, Analysis and Simulation of Distributed Systems*, DASD'03, Orlando, Florida, USA, 2003.

[32] Colin M. Ramsay. Exact waiting time and queue size distributions for equilibrium M/G/1 queues with pareto service. *Queueing Syst. Theory Appl.*, 57(4):147–155, December 2007.

[33] Burhaneddin Sandikci and Ihsan Sabuncuoglu. Analysis of the behavior of the transient period in non-terminating simulations. *European Journal of Operational Research*, 173(1):252–267, August 2006.

[34] L. Schruben, H. Singh, and L. Tierney. Optimal tests for initialization bias in simulation output. *Operations Research*, 31(6):1167–1178, 1983.

[35] Stephen C. Spratt. An evaluation of contemporary heuristics for the startup problem. Master's thesis, School of Engineering and Applied Science, University of Virginia, 1998.

[36] Peter D. Welch. The statistical analysis of simulation results. In *Computer Performance Modeling Handbook*, pages 268–328. Academic Press, New York, 1983.

[37] B. P. Welford. Note on a method for calculating corrected sums of squares and products. *Technometrics*, 4(3):pp. 419–420, 1962.

[38] K. Preston White, Jr. An effective truncation heuristic for bias reduction in simulation output. *Simulation*, 69:323–334, December 1997.

[39] K. Preston White, Jr. and Stewart Robinson. The problem of the initial transient (again), or why MSER works. In *Proceedings of the 2009 INFORMS Simulation Society Research Wrkshop*, pages 90–95, 2009.

[40] K. Preston White, Jr., Michael J. Cobb, and Stephen C. Spratt. A comparison of five steady-state truncation heuristics for simulation. In *Proceedings of the 32nd Conference on Winter Simulation*, WSC '00, pages 755–760, San Diego, CA, USA, 2000. Society for Computer Simulation International.

A Simulation Models

The range of simulation models used in this report are outlined below. With the exception of the M/Pareto/1 queueing model, this collection is based on the suite used by Eickhoff [5], which is referred to for further details.

A.1 M/G/1 Queueing Model

M/G/1 queues are simple queueing models with arrival times given by a Poisson process, service times given by a general distribution and a single server. Traffic intensity ρ is given by $\rho = \lambda/\mu$, for arrival rate λ and service rate μ. These queues are stable if and only if $\rho < 1$. All queueing models in this report used a service rate of $\mu = 1$, unless otherwise noted. Three measures of these queues are used:

- the *system state* (that is, number of customers in the system as seen by an arriving customer),
- *waiting time* of customers in the queue, and
- *response time* of customers in the system.

Initialisation bias and the initial transient phase can be influenced by adjusting the number of customers in the system when initialised. Note that queueing systems initialised with the steady-state mean number of customers can still have an initial transient phase, and initialising with a different number of customers can cause faster convergence to steady-state in some cases, [6].

A.1.1 M/M/1

M/M/1 queues are M/G/1 models with exponentially-distributed service times. Steady-state values for the mean system state θ_s, mean response time θ_r and mean waiting time θ_w are calculated as:

$$\theta_s = \frac{\rho}{1-\rho},$$
$$\theta_r = \frac{\rho}{\lambda(1-\rho)},$$
$$\theta_w = \frac{\rho}{\lambda(1-\rho)} - \rho/\lambda,$$

[14]. The respective steady-state CDFs for these parameters are given by

$$F_s(x) = \rho^x,$$
$$F_r(x) = 1 - e^{-x\mu(1-\rho)},$$
$$F_w(x) = 1 - \rho e^{-x\mu(1-\rho)}.$$

A.1.2 M/E_2/1

M/E_2/1 queues have service times governed by the Erlang distribution with shape parameter $k = 2$. The mean waiting time θ_w for such queues is given by

$$\theta_w = \frac{\rho - \rho^2/4}{1-\rho},$$

[14]. The waiting time CDFs were computed using a Maple program by McNickle, [23].

A.1.3 M/H_2/1

An M/H_2/1 queue has its service times distributed by the hyperexponential distribution. Its expected steady-state waiting time θ_w can be calculated as

$$\theta_w = \rho - \frac{\rho^2}{8(1-\rho)},$$

[14], and the waiting time CDFs were calculated with a Maple program by McNickle, [24].

A.1.4 M/Pareto/1

The Pareto distribution governs the service times of M/Pareto/1 queueing models, where the distribution has shape parameter α. The mean of the service time is only finite when $\alpha > 1$, and the variance only when $\alpha > 2$. The CDF for steady-state waiting times can be calculated using formulae presented by Ramsay [32], which also gives some specific points along this distribution for certain values of α.

A.2 Autoregressive Model

AR(1) autoregressive models have successive output values generated by the form

$$X_t = c + \varphi X_{t-1} + \varepsilon_t,$$

for constant value c and autoregressive parameter φ. The error term ε_t is Gaussian white noise with zero mean $E[\varepsilon_t] = 0$ and constant variance $Var[\varepsilon_t] = \sigma_\varepsilon^2$. All AR(1) processes in this report use $c = 0$ and $\sigma_\varepsilon^2 = 1$. As such, they have a steady-state distribution given by the standard normal distribution, with mean $E[X_\infty] = 0$ and variance $\sigma_X^2 = 1$.

Initialisation bias b is adjusted by varying the initial value X_{-1}, where $b = (X_{-1} - \theta)\sigma_X$, [29]. As $\theta = 0$ and $\sigma_X = 1$, this is simplified to $b = X_{-1}$.

A.3 Geometrical Autoregressive–Moving-Average Model

Autoregressive–moving-average (ARMA) models are a combination of autoregressive and moving average processes. An ARMA(p, q) process is given by the recurrence

$$X_t = c + \Psi_t + \sum_{i=1}^{q} \Theta_i \Psi_{t-i} + \sum_{i=1}^{p} \varphi_i X_{t-i},$$

where c is a constant, Ψ_t is a Gaussian white noise process with mean $E[\Psi_t] = 0$ and constant variance $Var[\Psi_t] = \sigma^2_{\Psi_t}$, Θ_i is the i^{th} moving average parameter, and φ_i is the i^{th} autoregressive parameter.

This report used geometrical ARMA(p, q) processes, where $p = q = k$ for some positive integer k and the parameters are defined by the $\frac{1}{2^i}$ geometrical series. This gives the form

$$X_t = 1 + \Psi_t + \sum_{i=1}^{k} \frac{1}{2^i}(X_{t-1} + \Psi_{t-1}).$$

These processes are initialised such that $X_{-k} = X_{-k+1} = \cdots = X_{-2} = X_{-1} = 0$.

The steady-state distributions for these geometrical ARMA(k, k) processes are given by a normal distribution. The first-order process ($k = 1$) has mean $E[X_\infty] = 2$ and variance $Var[X_\infty] = \frac{7}{3}$, and the second-order process ($k = 2$) has mean $E[X_\infty] = 4$ and variance $Var[X_\infty] = \frac{117}{25}$, [5].

A.4 Quadratic Displacement Process

A quadratic displacement process is Gaussian white noise with superimposed initial bias that quadratically converges to steady-state. It is given by

$$X_t = \begin{cases} \Psi_t + \frac{k}{l^2}(l-t)^2 & \text{for } t < l, \\ \Psi_t & \text{otherwise,} \end{cases}$$

where k is the initial offset and l is the length of the initial transient phase. Ψ_t is a Gaussian white noise process with zero mean $E[X_t] = 0$ and constant variance $Var[X_t] = 1$, and the steady-state distribution of the process is equivalent to this distribution.

A.5 Quadratic Stretch Process

A quadratic stretch process has a constant mean $E[X_i] = 0$, but its variability gradually increases during the initial transient phase until it approaches its steady-state value. It is described by

$$X_t = \begin{cases} (2i\frac{k}{l} - i^2\frac{k}{l^2})\Psi_t & \text{for } t < l, \\ k\Psi_t & \text{otherwise,} \end{cases}$$

where k is the eventual stretch during steady-state and l is the length of the initial transient phase (for the variance). The steady-state is normally-distributed with variance $Var[X_\infty] = k$.

A.6 Damped Vibration Process

A damped vibration process is an example of a process that converges non-monotonically to its steady-state value. The initial transient phase is given by an exponentially-diminishing oscillation superimposed over a Gaussian white noise process Ψ_t with zero mean $E[\Psi_t] = 0$ and variance $Var[\Psi_t] = 1$. Output value are generated by

$$X_t = \Psi_t + (ke^{i\frac{ln(0.05)}{l}}) \cdot \cos(\omega t),$$

where k is the initial amplitude of the vibration, $T = \frac{2\pi}{\omega}$ is the period length, and l is the length of the initial transient phase. The steady-state distribution is the standard normal distribution.

A.7 Random Walk Process

The random walk process uses a random walk R_t given by

$$R_t = \begin{cases} R_{t-1} + 1, & \text{with probability } 0.5, \\ R_{t-1} - 1, & \text{with probability } 0.5, \end{cases}$$

and is initialised with a value of $R_{-1} = 50$. The actual output value X_t is bounded to interval $[0, 100]$, outputting X_t as

$$X_t = \begin{cases} 0, & \text{for } R_t < 0, \\ R_t, & \text{for } 0 \leq R_t \leq 100, \\ 100, & \text{for } R_t > 100. \end{cases}$$

The steady-state distribution of this process is given by a discrete uniform distribution with $X_t = 0$ and $X_t = 100$ each having 0.5 probability. Thus, it has steady-state mean $E[X_\infty] = 50$ and variance $Var[X_\infty] = 2500$.

B Supplementary Figures

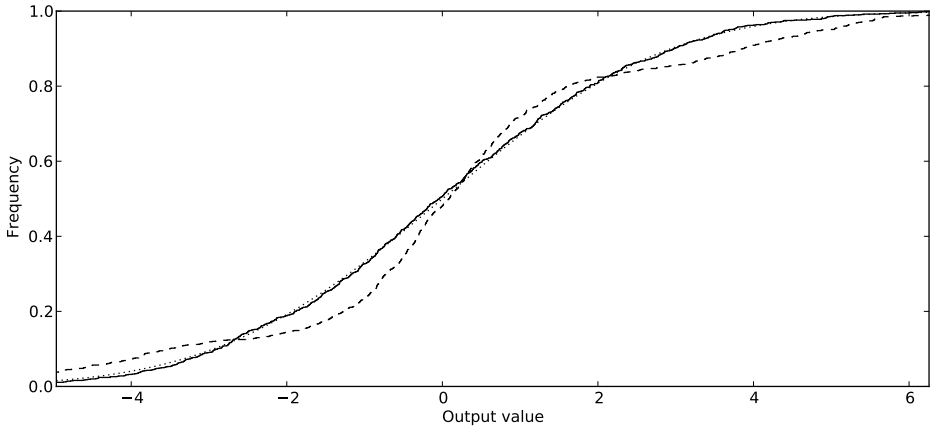


Figure B.1: AR(1) process with autoregressive parameter $\rho = 0.9$ and initial bias $b = 0$.

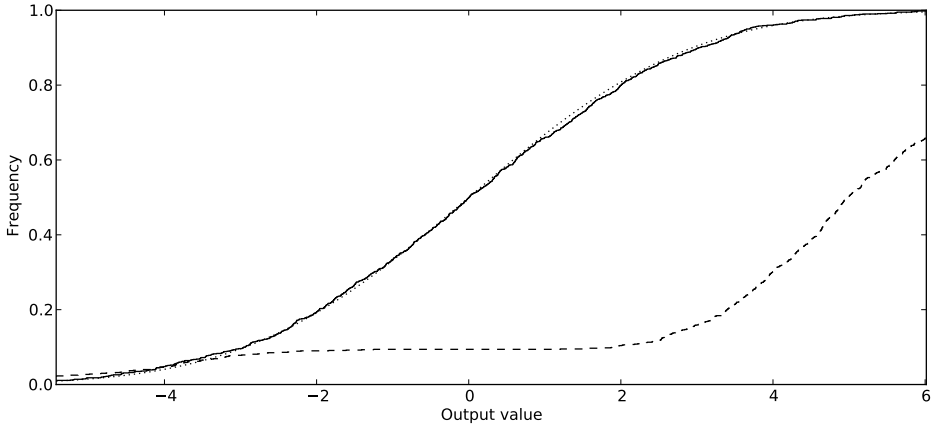


Figure B.2: AR(1) process with autoregressive parameter $\rho = 0.9$ and initial bias $b = 100$.

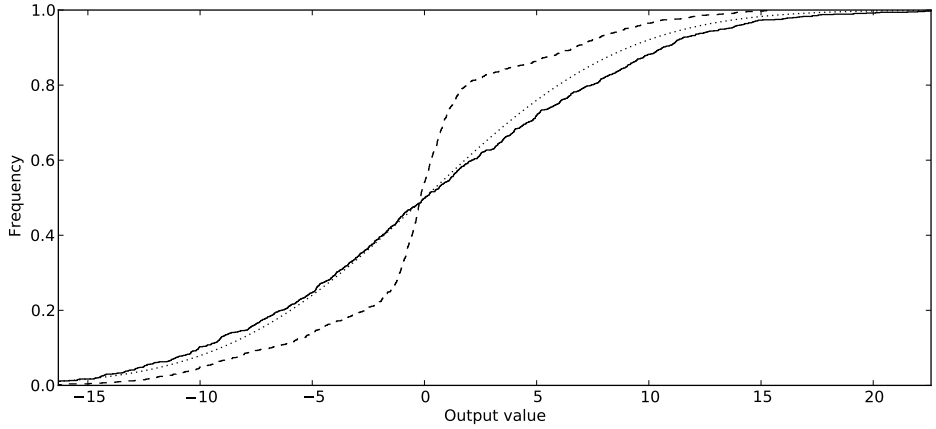


Figure B.3: AR(1) process with autoregressive parameter $\rho = 0.99$ and initial bias $b = 0$.

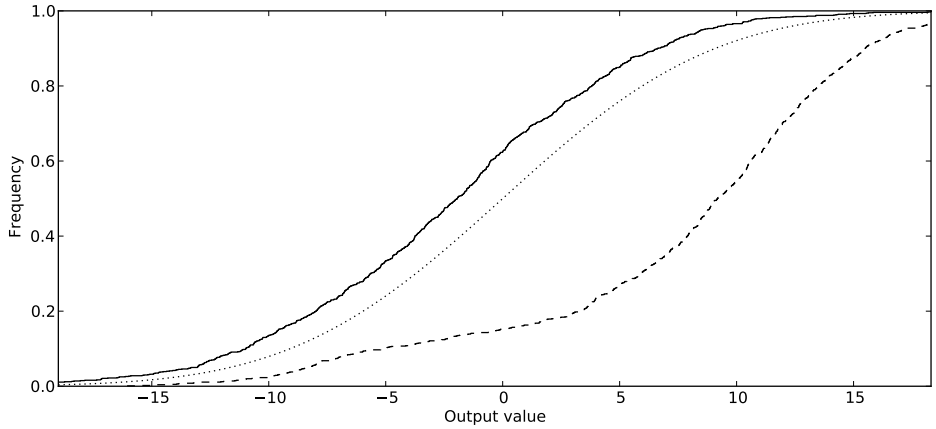


Figure B.4: AR(1) process with autoregressive parameter $\rho = 0.99$ and initial bias $b = 100$.

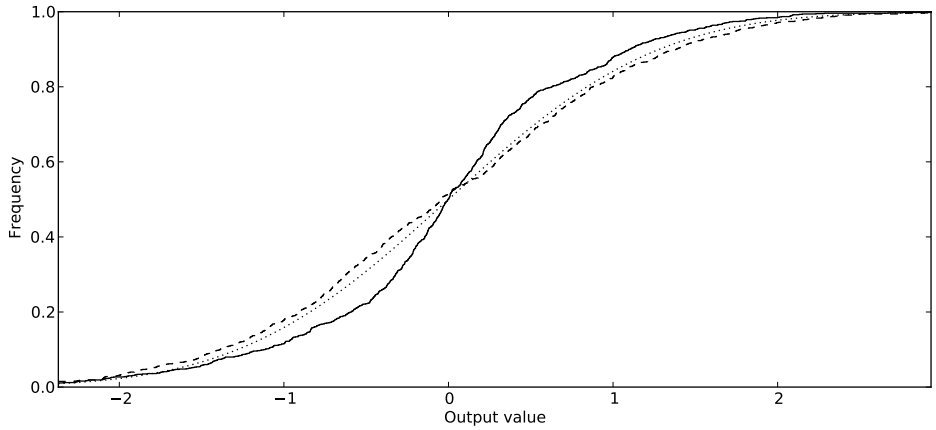


Figure B.5: AR(1) process with autoregressive parameter $\rho = 0$ and initial bias $b = 100$.

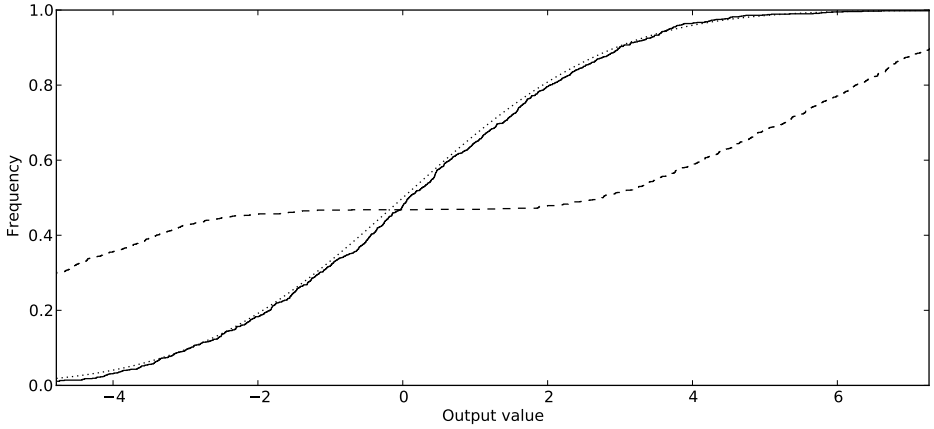


Figure B.6: AR(1) process with autoregressive parameter $\rho = -0.9$ and initial bias $b = 100$.

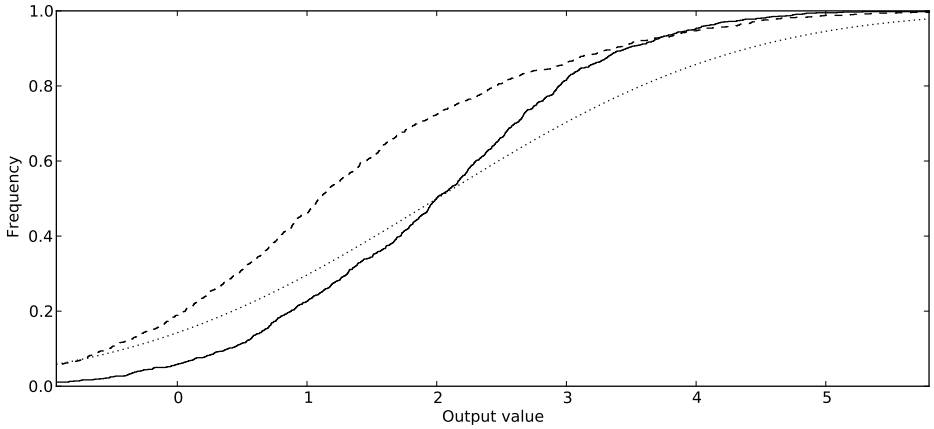


Figure B.7: Geometrical ARMA(1,1) process initialised with $X_{-1} = 0$.

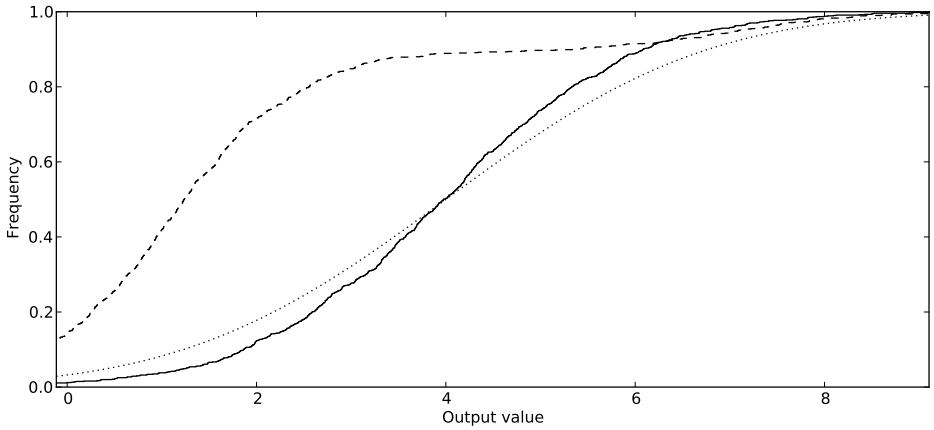


Figure B.8: Geometrical ARMA(2,2) process initialised with $X_{-2} = X_{-1} = 0$.

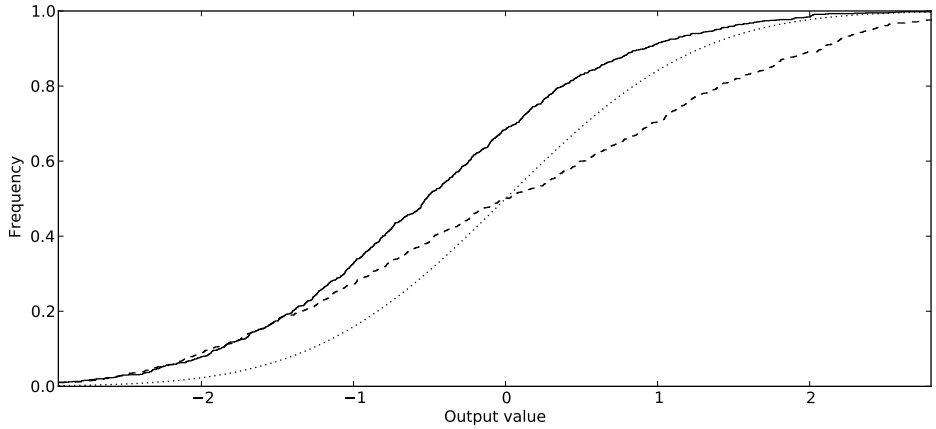


Figure B.9: Damped vibration process with amplitude $k = 10$, period $T = 50$, and transient length $l = 250$.

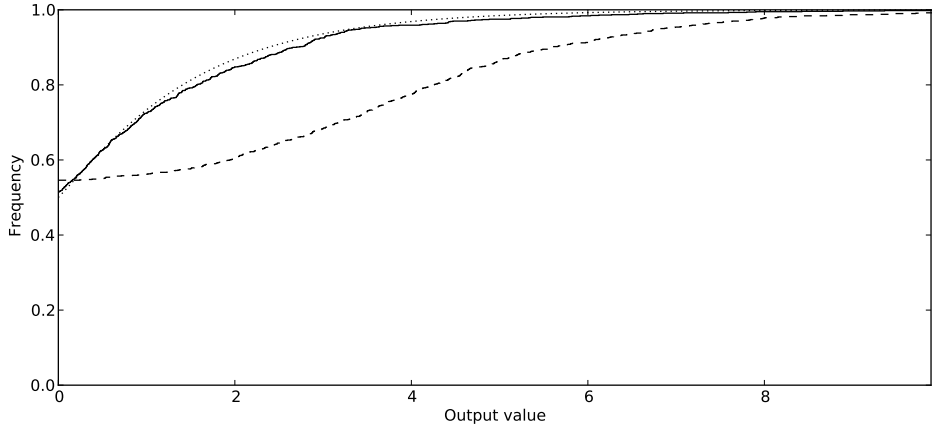


Figure B.10: Waiting times of an M/E_2/1 queue, with traffic intensity $\rho = 0.5$ and initialised empty and idle.

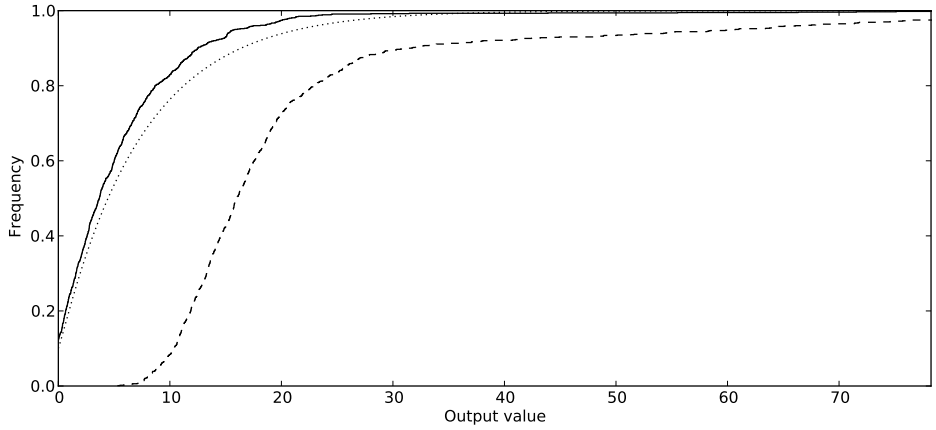


Figure B.11: Waiting times of an M/E_2/1, with traffic intensity $\rho = 0.9$ and 100 customers initially in the system.

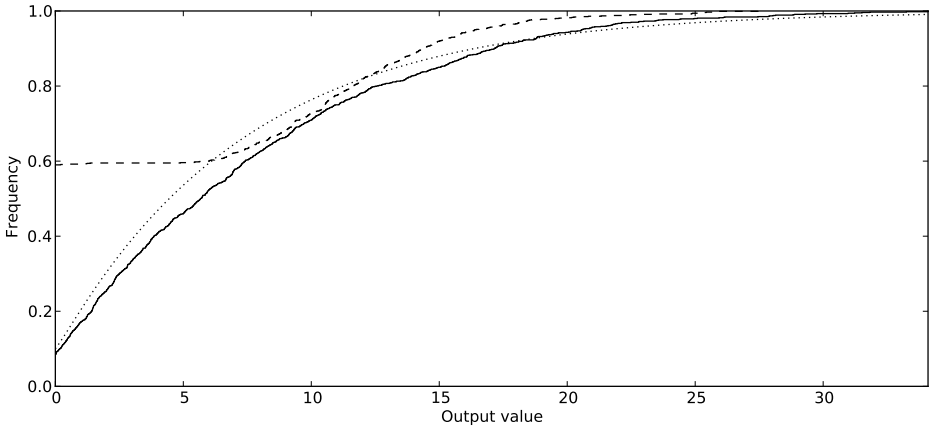


Figure B.12: Waiting times of an M/E_2/1 queue, with traffic intensity $\rho = 0.9$ and initialised empty and idle.

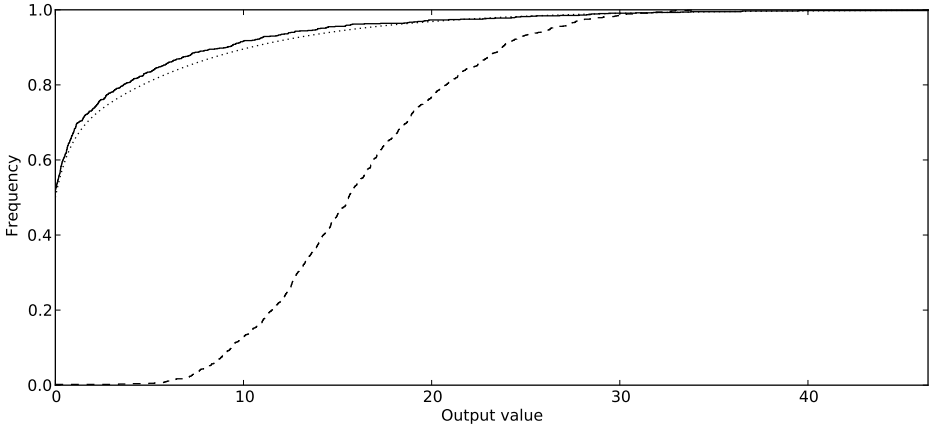


Figure B.13: Waiting times of an M/H_2/1 queue, with traffic intensity $\rho = 0.5$ and 100 customers initially in the system.

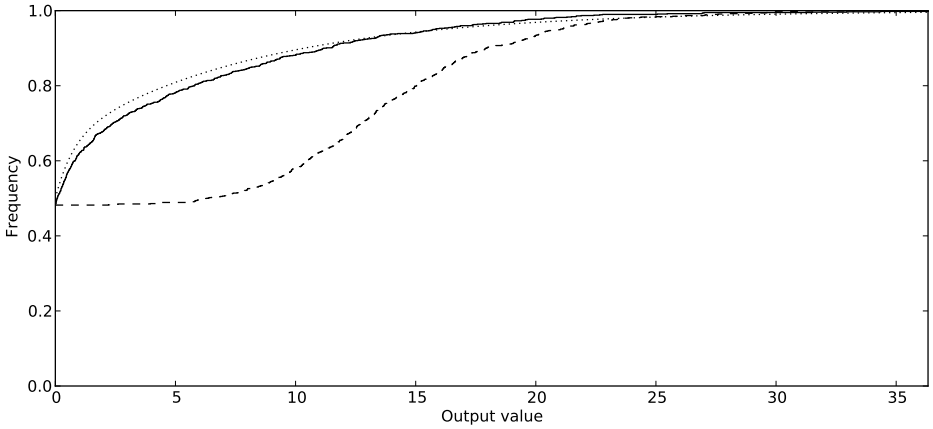


Figure B.14: Waiting times of an M/H_2/1 queue, with traffic intensity $\rho = 0.5$ and initialised empty and idle.

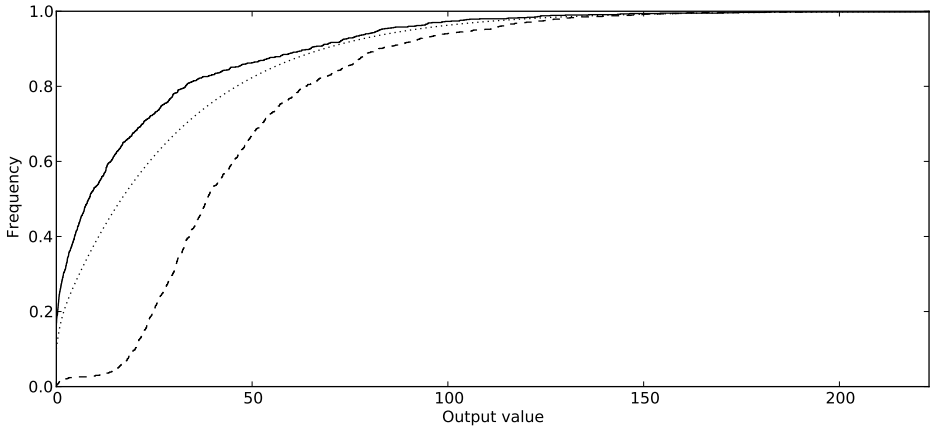


Figure B.15: Waiting times of an M/H_2/1 queue, with traffic intensity $\rho = 0.9$ and 100 customers initially in the system.

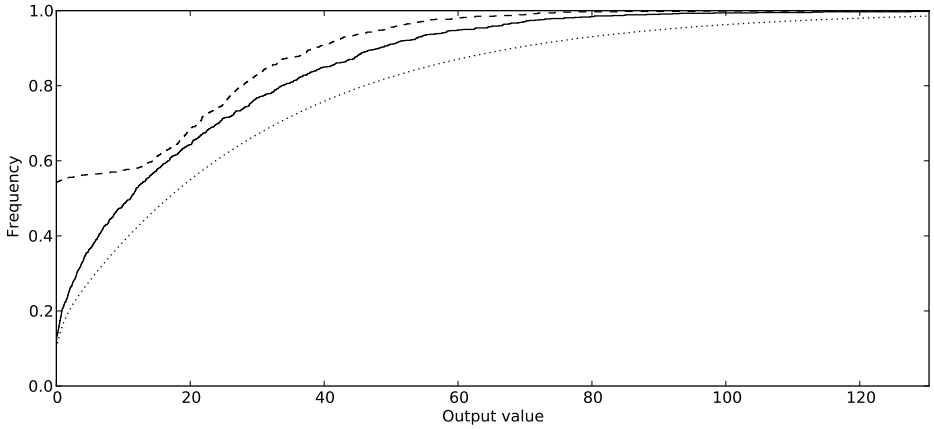


Figure B.16: Waiting times of an M/H_2/1 queue, with traffic intensity $\rho = 0.9$ and initialised empty and idle.

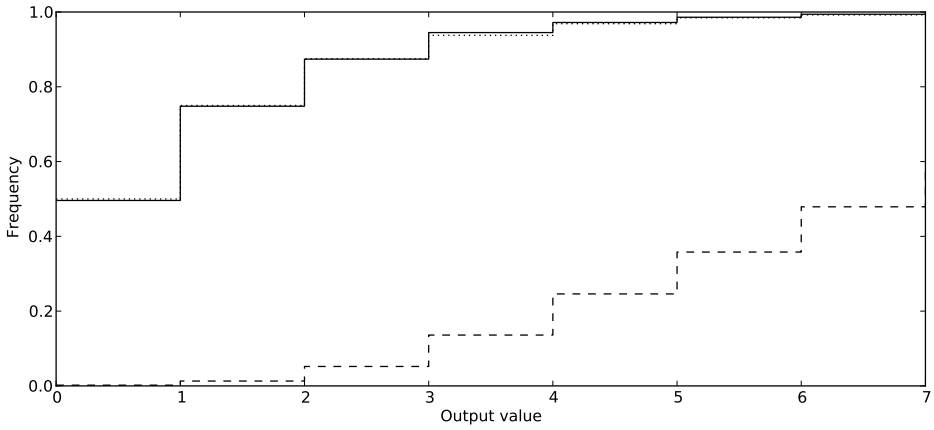


Figure B.17: System states of an M/M/1 queue, with traffic intensity $\rho = 0.5$ and 100 customers initially in the system.

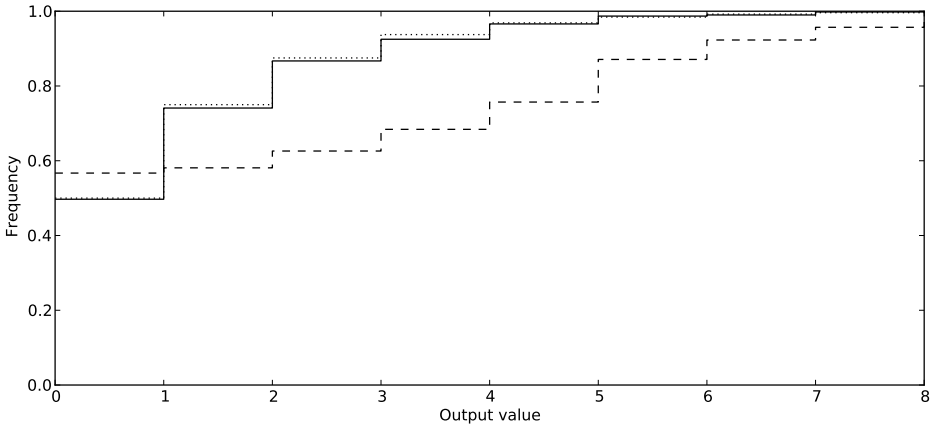


Figure B.18: System states of an M/M/1 queue, with traffic intensity $\rho = 0.5$ and initialised empty and idle.

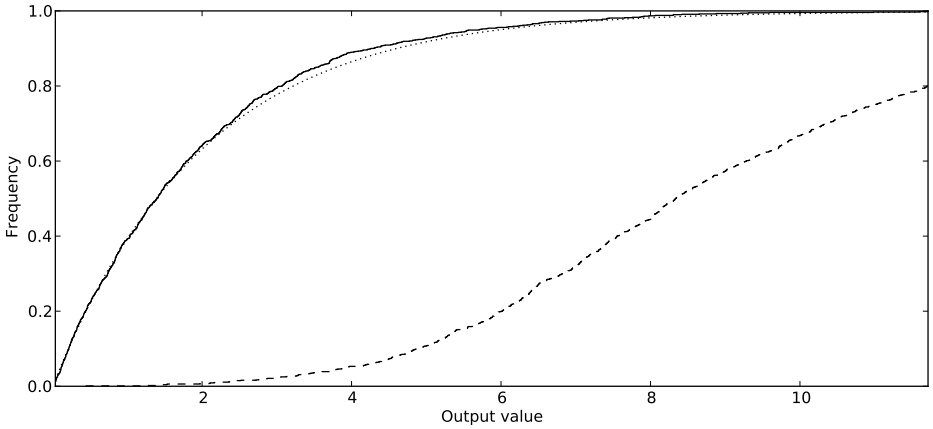


Figure B.19: Response times of an M/M/1 queue, with traffic intensity $\rho = 0.5$ and 100 customers initially in the system.

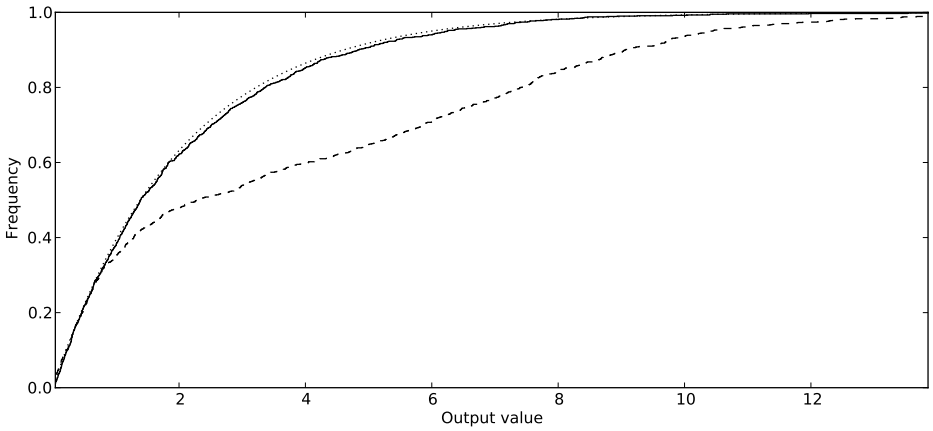


Figure B.20: Response times of an M/M/1 queue, with traffic intensity $\rho = 0.5$ and initialised empty and idle.

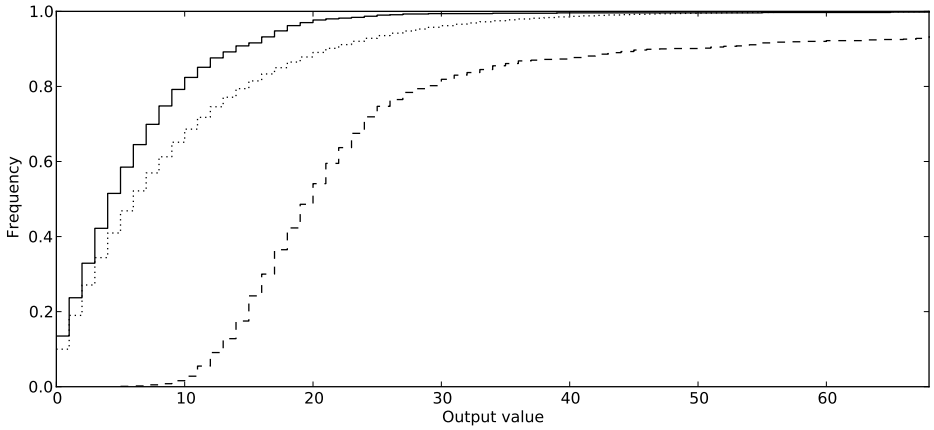


Figure B.21: System states of an M/M/1 queue, with traffic intensity $\rho = 0.9$ and 100 customers initially in the system.

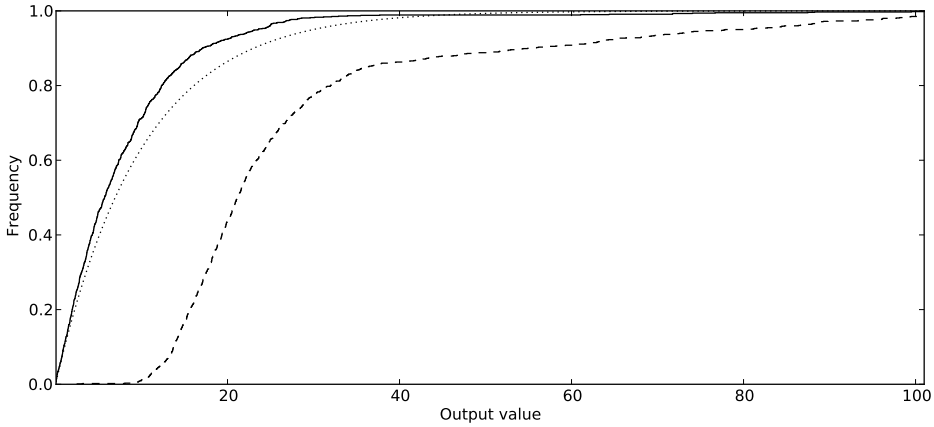


Figure B.22: Response times of an M/M/1 queue, with traffic intensity $\rho = 0.9$ and 100 customers initially in the system.

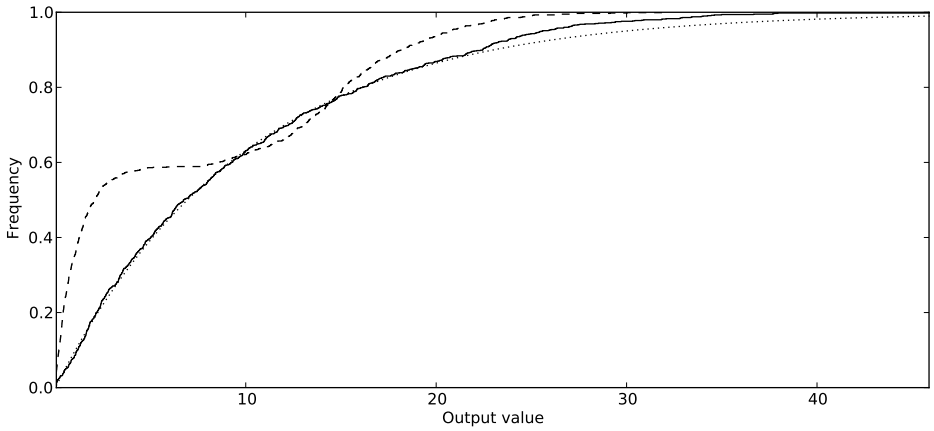


Figure B.23: Response times of an M/M/1 queue, with traffic intensity $\rho = 0.9$ and initialised empty and idle.

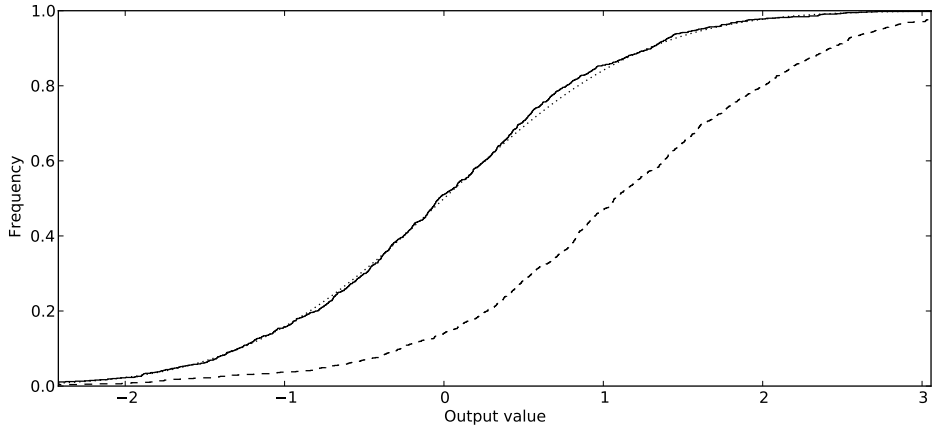


Figure B.24: Quadratic displacement process with initial displacement $k = 10$ and transient length $l = 100$.

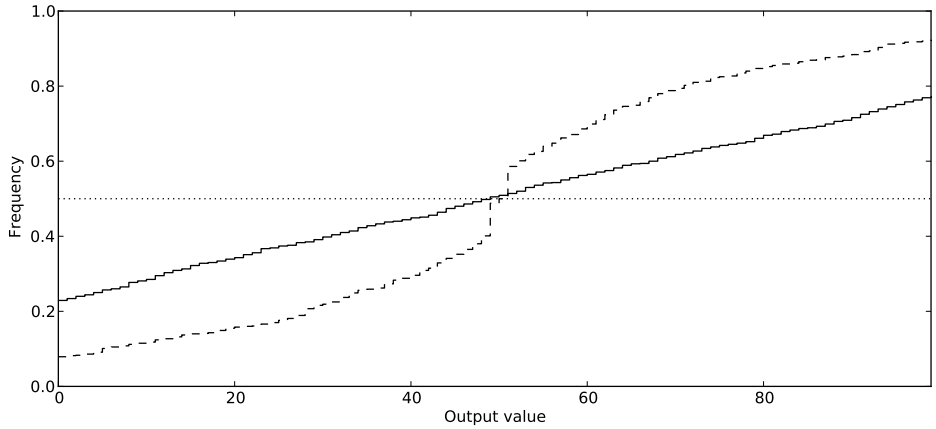


Figure B.25: Bounded random walk process initialised with $X_{-1} = 50$.

C Akaroa2 Implementation

For compilation with Akaroa2, the following two files are needed, and the path to their object file should be added to `AKANAL_OBJ` in *Mainfile.main*. `"CumulativeMeans"` also must be added to the list of possible transient methods in *variables.C* in Akaroa2.

C.1 cumulative_means_transient_detector.H

```
1  #ifndef cumulative_means_transient_detector_H
2  #define cumulative_means_transient_detector_H
3
4  #include <vector>
5  #include "../transient_detector.H"
6
7  class Environment;
8
9  class CumulativeMeansTransientDetector : public TransientDetector {
10   public:
11     CumulativeMeansTransientDetector(Environment* env);
12     virtual long TestObservations(long nobs, real obs[]);
13
14   protected:
15     Environment* environment;
16     // Number of observations in moving window, N
17     int windowSize;
18     // Smoothing factor
19     real alpha;
20     // Detection condition constant
21     real gamma;
22     // Current cumulative mean, C_t
23     real C;
24     // Current smoothed value
```

```
25      real s;
26      // Current test statistic, E_t
27      real E;
28      // Cumulative sum of observations
29      real cumSum;
30      // Buffer of one-step-ahead forecasting errors, e_t
31      std::vector<real> e;
32      // Current mean of forecasting errors
33      real eMean;
34      // M2 value for online calculation of variance
35      real eM2;
36  };
37
38  #endif
```

C.2 cumulative_means_transient_detector.C

```
1   #include <cmath>
2   #include <vector>
3   #include <iostream>
4
5   #include "akaroa/boolean.H"
6   #include "akaroa/real.H"
7   #include "../spectral/variance.H"
8   #include "environment.H"
9   #include "cumulative_means_transient_detector.H"
10
11  // Define this method for use in Akaroa2
12  DefineTransientDetectorType("CumulativeMeans",
        CumulativeMeansTransientDetector)
13
14  // Initalise variables, loading from the Akaroa environment
15  CumulativeMeansTransientDetector::CumulativeMeansTransientDetector(
        Environment *env) {
16      environment = env;
17      windowSize = env->GetInt("WindowSize");
18      alpha = env->GetReal("SmoothingFactor");
19      gamma = env->GetReal("DetectionMultiplier");
20      C = 0.0;
```

```
21          s = 0.0;
22          E = 0.0;
23          cumSum = 0.0;
24          eMean = 0.0;
25          eM2 = 0.0;
26      }
27
28      // Called whenever a new observation is collected for transient
            analysis
29      long CumulativeMeansTransientDetector::TestObservations(long nobs,
            real obs[]) {
30          // Get new observation
31          real ob = obs[nobs-1];
32          // Update cumulative sum
33          cumSum += ob;
34          // Calculate cumulative mean
35          C = cumSum / nobs;
36          if (nobs == 1) {
37              // Initial values for e_t and s
38              e.push_back(C);
39              s = C;
40          }
41          else {
42              // Obtain new forecasting error
43              e.push_back(s - C);
44              // Calculate next smoothed forecast
45              s = alpha * C + (1.0 - alpha) * s;
46          }
47          // Calculate M2 for online variance estimation
48          real delta = e.back() - eMean;
49          eMean += delta / e.size();
50          eM2 += delta * (e.back() - eMean);
51          // Update sliding window of sum of squared errors
52          E += pow(e.back(), 2.0);
53          if (nobs >= windowSize) {
54              if (nobs > windowSize) {
55                  E -= pow(e.at(nobs - windowSize - 1), 2.0);
56              }
```

```
57            // Detection condition: E <= gamma * N * S_e
58            if (E <= gamma * windowSize * sqrt(eM / (e.size() - 1))) {
59                // Truncation point found, return the point at the
                      start of the
60                // sliding window
61                return nobs - windowSize;
62            }
63        }
64        // No truncation point found yet
65        return -1;
66    }
```